

Министерство образования Российской Федерации
Государственное образовательное учреждение
высшего профессионального образования
"Челябинский государственный университет"

КОРПОРАТИВНЫЕ СИСТЕМЫ БАЗ ДАННЫХ
МЕТОДИЧЕСКИЕ УКАЗАНИЯ К ЛАБОРАТОРНЫМ РАБОТАМ

Челябинск 2003

Одобрено учебно-методическим советом математического факультета Челябинского государственного университета.

Методические указания содержат задание по разработке приложения модельной корпоративной базы данных, описание особенностей аппаратно-программной среды выполнения лабораторных работ и указания по созданию и настройке модельной корпоративной базы данных и разработке автоматизированных рабочих мест, предусмотренных заданием.

Методические указания предназначены для студентов 4 курса математического факультета (специальность 510200 "Прикладная математика и информатика").

Составитель канд. физ.-мат. наук, доц. М.Л. Цымблер

Рецензент канд. физ.-мат. наук, доц. Л.Б. Соколинский

Редактор Н.П. Мирдак

Подписано в печать 05.11.2003.

Формат 60×84 ¹/₁₆. Бумага газетная.

Печать офсетная. Уч.-изд. л. 2,0. Усл. печ. л. 1,7.

Тираж 50 экз. Заказ . Бесплатно

Челябинский государственный университет
454021 Челябинск, ул. Братьев Кашириных, 129

Полиграфический участок издательского центра ЧелГУ
454021 Челябинск, ул. Молодогвардейцев, 57^б

Введение

Целью выполнения лабораторных работ по курсу "Корпоративные системы баз данных" является *разработка приложения для модельной корпоративной базы данных* с использованием корпоративной СУБД Oracle и персональной СУБД MS Access.

Данное пособие содержит методические указания по выполнению лабораторных работ и состоит из трех разделов и списка литературы.

В *первом разделе* содержатся организационные указания по выполнению лабораторных работ (распределение ролей и работ при выполнении учебного задания командой студентов).

Во *втором разделе* содержится учебное задание на разработку приложения для модельной корпоративной базы данных.

В *третьем разделе* приводятся методические указания по выполнению лабораторных работ (установка необходимого программного обеспечения, создание и настройка модельной корпоративной базы данных и разработка АРМов, предусмотренных заданием).

В *списке литературы* даны книги, рекомендуемые для изучения при выполнении лабораторных работ.

1. Организационные указания

Учебное задание рассчитано на выполнение *коллективом из трех студентов* в течение 15-20 академических часов.

Разбиение на рабочие коллективы студенты группы проводят самостоятельно. Внутри коллектива студенты (также самостоятельно) выбирают *руководителя проекта*. В обязанности руководителя проекта, в частности, входит контроль и координация действий остальных участников проекта, взаимодействие с сотрудниками учебно-вычислительной лаборатории при решении технических проблем и отчетность за выполнение работ перед преподавателем, ведущим лабораторные занятия.

Распределение работ между участниками проекта и примерное время выполнения работ даны в табл. 1. Данная таблица используется руководителем проекта и преподавателем для промежуточного контроля при выполнении лабораторных работ.

2. Задание

Компания *Сложные Устройства для Бизнеса и Дома (СУБД)* занимается сборочным производством сложных устройств из деталей, закупаемых у поставщиков. В компании СУБД имеется *отдел поставок*, который занимается заказами деталей у поставщиков и учетом поставок деталей.

Требуется разработать *многопользовательскую систему учета поставок* для отдела поставок компании СУБД.

Отдел поставок располагает данными о деталях, поставщиках и поставках. Штатное расписание отдела предполагает должности Диспетчера, Экономиста и Начальника.

Таблица 1. План-график работ по выполнению задания

№ п/п	Вид работ	Распределение работ по участникам проекта			Время (час.)
		1	2	3	
1	Ознакомление с заданием				2
2	Установка и настройка программного обеспечения				
3	Создание рабочего пространства	Рабочее пространство	Рабочее пространство	Рабочее пространство	2
4	Создание таблиц	Таблица S	Таблица SP	Таблица P	
5	Создание индексов	Индексы таблицы S	Индексы таблицы SP	Индексы таблицы P	
6	Создание последовательностей	Последовательность s_seq	Последовательность p_seq	Последовательность sp_seq	
7	Добавление данных в таблицы	Таблица S	Таблица SP	Таблица P	
8	Создание представлений для реализации рабочих мест	Представления для Диспетчера	Представления для Экономиста	Представления для Начальника	2
9	Создание профилей пользователя	Профиль emp_profile	Профиль emp_profile	Профиль boss_profile	
10	Создание пользователей	Пользователь disp_user	Пользователь econ_user	Пользователь boss_user	
11	Создание синонимов имен пользователей	Синоним dispatcher	Синоним economist	Синоним boss	
12	Создание ролей и наделение их привилегиями	Роль disp_role	Роль econ_role	Роль boss_role	
13	Наделение пользователей привилегиями доступа к данным	Синоним dispatcher	Синоним economist	Синоним boss	
		Пользователь disp_user	Пользователь econ_user	Пользователь boss_user	
14	Разработка и тестирование триггеров	Уменьшение рейтинга поставщика в случае просроченной поставки	Запрет поставок высокотехнологичных и дорогих деталей от ненадежных поставщиков	Запрет поставки с весом более 1,5 т	3
		Увеличение рейтинга поставщика в случае заказа деталей	Модификация записей в необновляемом представлении Econ P		
15	Разработка и тестирование хранимых процедур	Вычисление превышения срока поставки	Вычисление количества просроченных поставок у поставщика	Вычисление общей суммы неустоек, предъявленных поставщику	3
		Вычисление суммы неустойки за просроченную поставку			
16	Разработка и тестирование АРМов	АРМ Диспетчера	АРМ Экономиста	АРМ Начальника	6
ИТОГО					18

Работа отдела поставок кратко может быть описана следующим образом. *Экономист* заказывает детали, которые необходимы компании. *Начальник* разрешает или отменяет заказ деталей. *Диспетчер* подготавливает путевые листы для автотранспорта компании, доставляющего детали от поставщиков, а в случае задержки поставки – требования об оплате неустойки.

2.1. Терминология

2.1.1. Рейтинг поставщика

Рейтинг поставщика – вещественное число, показатель надежности данного поставщика. Семантика значений рейтинга приводится в табл. 2.

Таблица 2. Семантика значений рейтинга поставщика

Значение	Семантика
Менее 6	Плохой
От 6 до 15	Посредственный
От 15 до 20 включительно	Приемлемый
Более 20	Хороший

При добавлении нового поставщика ему присваивается *рейтинг по умолчанию*, равный 15.

В процессе работы рейтинг поставщика изменяется следующим образом. В случае заказа деталей у поставщика его рейтинг увеличивается на 0,1. В случае просроченной поставки (см. 2.1.6) рейтинг поставщика уменьшается на 0,2.

2.1.2. Надежный и ненадежный поставщик

Поставщик, имеющий приемлемый или хороший рейтинг (см. табл. 2), является *надежным*, иначе поставщик является *ненадежным*.

2.1.3. Дорогая и дешевая деталь

Деталь с ценой более 1000 является *дорогой*, в противном случае деталь является *дешевой*.

Ненадежный поставщик НЕ может поставлять дорогую деталь. То есть Экономист не может заказать дорогую деталь у ненадежного поставщика. Вместе с тем возможна ситуация, когда ненадежный поставщик в прошлом был надежным и поставлял дорогие детали.

2.1.4. Высокотехнологичная деталь

Деталь может быть *продуктом высоких технологий*. Например, процессор является продуктом высоких технологий, а болт – нет.

Ненадежный поставщик НЕ может поставлять деталь, которая является продуктом высоких технологий. То есть Экономист не может заказать высокотехнологичную деталь у ненадежного поставщика. Вместе с тем возможна ситуация, когда ненадежный поставщик в прошлом был надежным и поставлял высокотехнологичные детали.

2.1.5. Состояние поставки

Поставка может находиться в одном из двух состояний: заказано или доставлено.

После того, как Экономист добавил запись о поставке, данная поставка считается *заказанной*. Начальник (и более никто) вправе удалить запись о заказанной поставке. Диспетчер подготавливает путевой лист для доставки заказанных деталей.

После того, как в записи о заказанной поставке Диспетчер заполнил дату доставки, данная поставка считается *доставленной*. Если превышен указанный Экономистом срок доставки, то Диспетчер подготавливает требование об оплате неустойки. Записи о доставленных поставках не подлежат удалению¹.

2.1.6. Просроченная поставка

Если превышен срок доставки², то данная поставка считается *просроченной*. Просроченные поставки учитываются при обновлении рейтинга поставщика (см. 2.1.1).

В случае просроченной поставки Диспетчер подготавливает требование об оплате поставщиком *неустойки*. Сумма неустойки вычисляется как 0,1% от суммы поставки за каждый просроченный день, но не может превышать 10% от суммы поставки.

2.2. Краткая постановка задачи

Требуется разработать *многопользовательскую систему учета поставок* для отдела поставок компании СУБД. Основные функции системы:

- Организация базы данных в соответствии со структурой, описанной в разделе 2.3.
- Обеспечение целостности базы данных в соответствии с разделом 2.4.
- Обеспечение многопользовательского режима работы автоматизированных рабочих мест в соответствии с требованиями, описанными в разделе 2.5.

2.3. Описание таблиц

При описании таблиц базы данных используются обозначения, указанные в табл. 3.

Таблица 3. Обозначения ключевых полей таблиц

Обозначение	Семантика
*	Поле является первичным ключом
^Таблица.Поле	Поле является внешним ключом, ссылающимся на указанное поле указанной таблицы

¹ Мы считаем, что далее данные о поставках используются в складском учете компании СУБД.

² Мы считаем, что в превышении сроков доставки виноват поставщик, а не автотранспорт компании СУБД.

2.3.1. Поставщики (S)

№ п/п	Поле	Ключ	Семантика
17	S#	*	Уникальный код поставщика
18	SName		Имя поставщика
19	SCity		Город поставщика
20	Address		Почтовый адрес поставщика
21	Rating		Рейтинг поставщика (см. 2.1.1)

2.3.2. Детали (P)

№ п/п	Поле	Ключ	Семантика
1	P#	*	Уникальный код детали
2	PName		Имя детали
3	HTP		Является ли деталь продуктом высоких технологий, High Technology Product (см. 2.1.4)
4	Weight		Вес детали <i>в граммах</i>

2.3.3. Поставки (SP)

№ п/п	Поле	Ключ	Семантика
1	SP#	*	Уникальный код поставки
2	S#	^S.S#	Уникальный код поставщика
3	P#	^P.P#	Уникальный код детали
4	Qty		Количество деталей в поставке
5	Price		Цена одной детали
6	OrderDate		Дата заказа поставки
7	Period		Срок доставки в днях
8	ShipDate		Дата доставки

2.4. Ограничения целостности

2.4.1. Целостность полей

Таблица	Поле	Ограничение целостности
S	SName	NOT NULL
	Scity	NOT NULL
	Address	NOT NULL
	SName, Scity, Address	UNIQUE
P	PName	NOT NULL
	Weight	>0
	HTP	in (0, 1)
	PName, Weight, HTP	UNIQUE
SP	OrderDate	NOT NULL
	Qty	>0
	Price	>0
	Period	>=0

2.4.2. Целостность таблиц

Таблица	Ограничение целостности
S	Нет
P	Нет
SP	OrderDate<=ShipDate

2.4.3. Целостность по ссылкам

Обновление/удаление объекта ссылки внешнего ключа вызывает каскадное обновление/удаление соответствующих записей.

2.4.4. Целостность базы данных

1. Ненадежный поставщик не может поставлять дорогие и/или высокотехнологичные детали.
2. Вес поставки не должен превышать 1,5 тонн.

2.5. Описание рабочих мест

2.5.1. Диспетчер

2.5.1.1. Представление данных

Представление Поставщика (Disp_S)

№ п/п	Поле	Семантика
1	S#	Уникальный код поставщика
2	SName	Имя поставщика
3	SCity	Город поставщика
4	Address	Почтовый адрес поставщика

Примечание. Записи в таблице Disp_S сортируются по полям SName и SCity (в алфавитном порядке).

Представление Поставки (Disp_SP)

№ п/п	Поле	Семантика
1	SP#	Уникальный код поставки
2	SName	Имя поставщика
3	PName	Имя детали
4	Sum	Сумма поставки
5	Weight	Вес <i>поставки</i> , в килограммах
6	OrderDate	Дата заказа поставки
7	Period	Срок доставки в днях
8	ShipDate	Дата доставки
9	Status	Состояние поставки (см. 2.1.5) <i>словами</i>
10	IsOverdue	Просрочена ли поставка (см. 2.1.6) <i>словами</i>
11	SCity	Город поставщика
12	Address	Адрес поставщика

Примечание. В таблице Disp_SP должны отображаться записи только о заказанных (см. 2.1.5) и просроченных (см. 2.1.6) поставках. Записи в таблице Disp_SP сортируются по убыванию значения полей Sum и Weight, далее – по полям PName и SName (в алфавитном порядке) и группируются по значению поля Status.

Представление Доставка (Disp_Shipment)

№ п/п	Поле	Семантика
1	SP#	Уникальный код поставки
2	ShipDate	Дата доставки

Примечание. В таблице Disp_Shipment должны отображаться записи только о заказанных поставках (см. 2.1.5).

2.5.1.2. Доступ к данным

Таблица/представление	Операция			
	SELECT	INSERT	UPDATE	DELETE
S	-	-	-	-
P	-	-	-	-
SP	-	-	-	-
Disp_S	+	+	+	-
Disp_SP	+	-	-	-
Disp_Shipment	+	-	+	-

2.5.1.3. Отчеты

1. Путевой лист на заданную поставку.
2. Требование об оплате неустойки за заданную просроченную поставку.
3. Список просроченных поставок за заданный период.

2.5.1.4. Интерфейс пользователя

Главная форма



Форма "Поставщики"

The 'Suppliers' form is a rectangular window with a blue border. It features four input fields stacked vertically, labeled 'Код', 'Имя', 'Город', and 'Адрес'. To the right of these fields is a vertical scrollbar. Below the input fields is a button labeled 'Добавить поставщика...'. The form has a light gray background and a 3D effect.

Форма "Поставки"

Код	Поставщик	Деталь	Сумма	Вес (кг)	Дата заказа	Срок доставки (дней)	Дата доставки	Состояние	Просрочена?	Город	Адрес

Обновить Путевой лист... Неустойка... Просроченные...

Форма "Доставка"

Код поставки	Дата доставки

Обновить

Форма "О программе..."

Автоматизированное рабочее место
ДИСПЕТЧЕРА отдела поставок

© 2002 И.О. Фамилия разработчика
email для контактов

ОК

2.5.2. ЭКОНОМИСТ

2.5.2.1. Представление данных

Представление Поставщика (Econ_S)

№ п/п	Поле	Семантика
1	S#	Уникальный код поставщика
2	SName	Имя поставщика
3	SCity	Город поставщика
4	Rating	Рейтинг поставщика
5	RatingWord	Рейтинг поставщика <i>словами</i> (см. табл. 2)

Примечание. Записи в таблице Econ_S сортируются по убыванию значения поля Rating, далее – по полю SName (в алфавитном порядке) и группируются по значению поля SCity.

Представление Детали (Econ_P)

№ п/п	Поле	Семантика
1	P#	Уникальный код детали
2	PName	Имя детали
3	IsHTP	Является ли деталь продуктом высоких технологий (см. 2.1.4) <i>словами</i>
4	Weight	Вес детали <i>в граммах</i>

Примечание. Записи в таблице Econ_P сортируются по полю PName (в алфавитном порядке), далее – по убыванию значения поля Weight и группируются по значению поля IsHTP.

Представление Поставки (Econ_SP)

№ п/п	Поле	Семантика
1	S#	Уникальный код поставщика
2	P#	Уникальный код детали
3	SName	Имя поставщика
4	PName	Имя детали
5	IsExpensive	Является ли деталь дорогой или дешевой (см. 2.1.3) <i>словами</i>
6	IsHTP	Является ли деталь продуктом высоких технологий (см. 2.1.4) <i>словами</i>
7	Qty	Количество деталей в поставке
8	Price	Цена одной детали
9	Sum	Сумма поставки
10	Weight	Вес <i>поставки в килограммах</i>
11	Period	Срок поставки в днях

Примечание. В таблице Econ_SP должны отображаться записи только о заказанных поставках (см. 2.1.5). Записи в таблице Econ_SP сортируются по убыванию значения полей Sum, Weight и по возрастанию значения поля Period, далее – по полям PName и SName (в алфавитном порядке).

2.5.2.2. Доступ к данным

Таблица/представление	Операция			
	SELECT	INSERT	UPDATE	DELETE
S	-	-	-	-
P	-	-	-	-
SP	-	+	-	-
Econ_S	+	-	-	-
Econ_P	+	+	+	-
Econ_SP	+	-	-	-

2.5.2.3. Отчеты

1. Список надежных/ненадежных поставщиков, упорядоченный по убыванию рейтинга и сгруппированный по городу поставщика.
2. Список дорогих/дешевых деталей, упорядоченный по убыванию цены и возрастанию веса детали и сгруппированный по технологичности детали.
3. Список высокотехнологичных/низкотехнологичных деталей, упорядоченный по убыванию цены и возрастанию веса детали.

2.5.2.4. Интерфейс пользователя

Главная форма

Форма "Поставщики"

Код	Поставщик	Город	Рейтинг

Форма "Детали"

Код	Деталь	Высокие технологии?	Вес

Форма "Поставки"

Код поставщика	Код детали	Поставщик	Деталь	Дорогая/дешевая	Высокие/низкие технологии	Кол-во	Цена	Сумма	Вес (кг)	Срок доставки (дней)	Дата доставки

Форма "О программе..."

Имеет интерфейс, аналогичный форме "О программе" АРМа диспетчера (см. 2.5.1.4).

2.5.3. Начальник

2.5.3.1. Представление данных

Представление Поставщика (Boss_S)

№ п/п	Поле	Семантика
1	S#	Уникальный код поставщика
2	SName	Имя поставщика

№ п/п	Поле	Семантика
3	Rating	Рейтинг поставщика
4	SCity	Город поставщика
5	Address	Почтовый адрес поставщика

Примечание. Записи в таблице Boss_S сортируются по убыванию значения поля Rating, далее – по полю SName (в алфавитном порядке) и группируются по значению поля SCity.

Представление Детали (Boss_P)

№ п/п	Поле	Семантика
1	P#	Уникальный код детали
2	PName	Имя детали
3	IsHTP	Является ли деталь продуктом высоких технологий (см. 2.1.4) <i>словами</i>
4	Weight	Вес детали <i>в граммах</i>

Примечание. Записи в таблице Boss_P сортируются по полю PName (в алфавитном порядке), далее – по убыванию значения поля Weight и группируются по значению поля IsHTP.

Представление Поставки (Boss_SP)

№ п/п	Поле	Семантика
1	SP#	Уникальный код поставки
2	SName	Имя поставщика
3	PName	Имя детали
4	IsExpensive	Является ли деталь дорогой или дешевой (см. 2.1.3) <i>словами</i>
5	IsHTP	Является ли деталь продуктом высоких технологий (см. 2.1.4) <i>словами</i>
6	Sum	Сумма поставки
7	SCity	Город поставщика
8	OrderDate	Дата заказа поставки
9	Period	Срок доставки в днях
10	ShipDate	Дата доставки
11	Status	Состояние поставки (см. 2.1.5) <i>словами</i>
12	IsOverdue	Просрочена ли поставка (см. 2.1.6) <i>словами</i>

Примечание. Записи в таблице Boss_SP сортируются по возрастанию значения поля OrderDate, далее – по убыванию значения поля ShipDate и группируются по значению поля Status.

2.5.3.2. Доступ к данным

Таблица/представление	Операция			
	SELECT	INSERT	UPDATE	DELETE
S	+	+	+	-
P	+	+	+	-
SP	+	+	+	+
Boss_S	+	+	+	-
Boss_P	+	+	+	-
Boss_SP	+	-	-	-

2.5.3.3. Отчеты

1. Список поставщиков с указанием суммы их поставок за заданный период.
2. Список деталей с указанием суммы их поставок за заданный период.
3. Список поставщиков с указанием количества просроченных ими поставок и общей суммы предъявленных им неустоек за заданный период.

2.5.3.4. Интерфейс пользователя

Главная форма

Форма "Поставщики"

Форма "Детали"

Форма "Поставки"

Форма "О программе..."

Имеет интерфейс, аналогичный форме "О программе" АРМа диспетчера (см. 2.5.1.4).

2.6. Приложение. Образцы выходных документов

2.6.1. Путевой лист

Компания "СУБД" Отдел поставок Путевой лист от «Дата заказа поставки»
Описание поставки: Деталь: «Имя детали» (код «Уникальный код детали») Поставщик: «Имя поставщика» (код «Уникальный код поставщика») Количество: «Количество деталей в поставке» Вес нетто: «Общий вес деталей в поставке в килограммах» кг Ценность: «Сумма поставки» руб.
Адрес доставки: Город: «Город поставщика» Адрес: «Адрес поставщика»

2.6.2. Требование об оплате неустойки

Компания "СУБД" Отдел поставок «Дата доставки»	«Город поставщика» «Адрес поставщика» «Имя поставщика»
--	--

Уважаемые господа,

Настоящим письмом сообщаем Вам, что по вине Вашей компании поставка детали «Имя детали» в количестве «Количество деталей в поставке» шт. от «Дата заказа поставки» просрочена на «Превышение срока поставки в днях» дней.

В соответствии действующим законодательством просим Вас незамедлительно выплатить нашей компании неустойку в размере «Сумма неустойки» руб.

Начальник отдела поставок

3. Методические указания

Данный раздел содержит методические указания по выполнению учебного задания и состоит из трех частей.

В *первой части* приводится описание особенностей аппаратно-программной среды выполнения лабораторных работ и указания по установке необходимого программного обеспечения.

Во *второй части* содержатся указания по созданию и настройке модельной корпоративной базы данных.

В *третьей части* приводятся указания по разработке АРМов (автоматизированных рабочих мест), предусмотренных заданием.

3.1. Технический контекст лабораторных работ

В ходе выполнения лабораторных работ Вам предстоит разработать приложение для модельной корпоративной базы данных. При этом Вы будете использовать следующее аппаратно-программное обеспечение: на сервере локальной сети установлена СУБД Oracle Server [1; 2; 3; 4; 6; 7], на рабочих станциях установлена ОС Windows и СУБД MS Access [5].

Предварительно получите у преподавателя, ведущего лабораторные занятия, инструкции по установке необходимых для работы программных продуктов.

Техническое подключение к серверу СУБД Oracle с рабочей станции обеспечивается *драйвером SQL*Net*, который является продуктом фирмы Oracle. Установите на рабочей станции драйвер SQL*Net, входящий в состав пакета Oracle for Windows .

Доступ к таблицам базы данных на сервере СУБД Oracle с помощью запроса SQL из СУБД, установленной на рабочей станции, обеспечивается *драйвером ODBC (Open Database Connectivity)*. Драйвер ODBC может быть продуктом фирмы Oracle или фирмы-производителя СУБД. Установите на рабочей станции драйвер Microsoft ODBC for Oracle.

Разработка приложения базы данных осуществляется с помощью *инструментальной системы*. Фирма Oracle предоставляет разработчику мощные инструментальные системы Oracle Developer и Oracle Designer [2], которые обеспечивают автоматическое проектирование и нормализацию структуры базы данных и поддерживают все этапы технологического цикла разработки приложения базы данных. Поэтому данные инструментальные системы наиболее часто используются для разработки сложных приложений для крупных корпоративных баз данных.

При разработке модельного приложения Вам достаточно будет использовать менее сложную инструментальную систему *PL/SQL Developer* фирмы Allround Automations [<http://www.allroundautomations.nl/bodyplsqldev.html>]. Данный программный продукт обеспечивает интегрированную среду разработки, включающую в себя текстовый редактор, компилятор и отладчик программного кода, навигатор объектов схемы и др. Установите на рабочей станции приложение PL/SQL Developer.

При выполнении указаний последующих разделов Вы будете выступать в роли администратора сервера базы данных и разработчика клиентских приложений. В соответствии заданием Вам необходимо будет подключаться к серверу СУБД Oracle в качестве следующих пользователей сервера: Администратор, Диспетчер, Экономист, Начальник. Узнайте у преподавателя, ведущего лабораторные занятия, имена и пароли соответствующих учетных записей. Далее в целях удобства изложения мы будем использовать следующие

имена учетных записей этих пользователей: `admin_user`, `disp_user`, `econ_user`, `boss_user`.

В настоящее время по ряду очевидных причин обеспечение каждого студента отдельным сервером с установленной СУБД Oracle Server не является возможным. Поэтому пользователь `admin_user` не является полноценным администратором сервера СУБД Oracle. В частности, пользователь `admin_user` не может создавать табличные пространства, пользователей и роли.

При выполнении указаний дальнейших разделов Вам потребуется следующая документация по СУБД Oracle [<http://otn.oracle.com/documentation/>]:

- Server Concepts Manual – терминология, основные возможности и средства СУБД Oracle;
- Administrators Guide – руководство администратора базы данных Oracle;
- Application Developers Guide – руководство разработчика приложений базы данных Oracle;
- PL/SQL Users Guide and Reference – руководство и справочник по языку PL/SQL;
- SQL Language Reference Manual – руководство и справочник по языку SQL.

При возникновении в ходе выполнения указаний теоретической проблемы рекомендуется сначала прочитать соответствующий раздел документа Server Concepts Manual, а затем, в случае необходимости и в зависимости от контекста проблемы – остальные документы.

Узнайте у преподавателя, ведущего лабораторные занятия, как получить документацию по СУБД Oracle.

3.2. Администрирование базы данных

При выполнении указаний данного раздела Вы выступаете в роли администратора базы данных Oracle. *Администратор базы данных* – наиболее привилегированный пользователь, управляющий базой данных. Одними из основных обязанностей администратора являются заведение пользователей, управление физической и логической структурой базы данных.

Указания данного раздела необходимо выполнять, подключившись к базе данных Oracle как пользователь `admin_user`.

Создайте текстовый файл `corp_dbms.sql`. В него необходимо последовательно *записывать команды PL/SQL, предусмотренные указаниями*. Каждая команда должна предваряться соответствующим *комментарием-спецификацией*.

В данном разделе имеются указания, корректное выполнение которых требует наличия прав, которыми пользователь `admin_user` не обладает. Такие указания помечены символом ☒. Выполнение команд, предусмотренных данными указаниями, *не требуется*, текст команд в файле `corp_dbms.sql` необходимо закомментировать. *Вместо* данных команд

необходимо выполнить команды, предусмотренные указанием, которое помечено символом .

Операции создания и настройки объектов модельной корпоративной базы данных (таблицы, представления, пользователи и др.) необходимо *тестировать*. При тестировании следует использовать в том числе словарь базы данных Oracle.

Словарь базы данных представляет собой набор таблиц, в которых хранится различная системная информация о базе данных (количество и имена таблиц и представлений, структура таблиц и др.). Все пользователи базы данных, за исключением администратора, имеют доступ к словарю данных только по чтению.

Словарь базы данных Oracle содержит три набора таблиц, имеющих одинаковую структуру и различающихся префиксом имени:

- `user_` – объекты базы данных, созданные пользователем;
- `all_` – все объекты базы данных, к которым имеет доступ пользователь;
- `dba_` – все объекты базы данных, к которым имеют доступ все пользователи.

Используя документацию по СУБД Oracle, изучите информацию о таблице `user_objects` словаря базы данных.

3.2.1. Создание табличного пространства

Табличное пространство – логическая единица хранения базы данных Oracle. Администратор базы данных использует табличное пространство для группировки объектов одного приложения (таблиц, индексов, представлений, снимков и др.) для упрощения некоторых административных операций (например, резервное копирование).

Создайте для разрабатываемого приложения модельной корпоративной базы данных табличное пространство с именем `DBMS`. Используйте команду `PL/SQL CREATE TABLESPACE`.

Используя документацию по СУБД Oracle, изучите информацию о таблице `user_tablespaces` словаря базы данных. Протестируйте создание табличного пространства с помощью следующего запроса к словарю данных:

```
SELECT tablespace_name, status  
FROM user_tablespaces
```

3.2.2. Создание таблиц

Таблица – основная единица хранения данных в базе данных Oracle.

Создайте описанные в 2.3 таблицы модельной корпоративной базы данных. Используйте команду `PL/SQL CREATE TABLE`.

Указанные в задании ограничения целостности полей и таблиц и ограничения целостности по ссылкам реализуйте как *именованные* ограничения целостности. Используйте общепринятые префиксы имен ограничений целостности, указанные в табл. 4.

Таблица 4. Именованние ограничений целостности таблиц

Префикс имени	Семантика ограничения	Пример ограничения
pk_	Ограничение первичного ключа таблицы (PRIMARY KEY)	s# number constraint pk_sid primary key
fk_	Ограничение внешнего ключа таблицы (FOREIGN KEY)	constraint fk_sid foreign key (s#) references s (s#) on delete cascade
nn_	Запрет пустых значений в столбце таблицы (NOT NULL)	sname char(20) constraint nn_sname NOT NULL
unq_	Запрет совпадающих значений в столбце таблицы (UNIQUE)	constraint unq_namecityaddr unique (sname, scity, address)
chk_	Запрет значений в столбце таблицы, нарушающих указанное условие (CHECK)	weight number constraint chk_weight (weight>0)

Используя документацию по СУБД Oracle, изучите информацию о таблицах словаря базы данных `user_tab_columns` и `user_constraints`. Протестируйте создание таблиц с помощью следующих запросов к словарю данных:

- ```
SELECT object_name, object_type
FROM user_objects
```
- ```
SELECT table_name, column_name, data_type,
data_length,
nullable, data_default
FROM user_tab_columns
ORDER BY table_name
```
- ```
SELECT table_name, constraint_name, constraint_type,
delete_rule, status
FROM user_constraints
ORDER BY table_name, constraint_name
```

### 3.2.3. Создание индексов

*Индекс* – объект схемы для быстрого поиска данных в таблицах/представлениях.

Проанализируйте, по каким полям необходимо упорядочивать записи в представлениях и отчетах. Создайте для соответствующих таблиц соответствующие индексы (обратите внимание, когда нужно создать индекс не по одному полю, а по комбинации полей). Используйте команду PL/SQL `CREATE INDEX`.

Используя документацию по СУБД Oracle, изучите информацию о таблицах словаря базы данных `user_indexes` и `user_ind_columns`. Протестируйте создание индексов с помощью следующих запросов к словарю данных:

- `SELECT index_name, table_name, uniqueness  
FROM user_indexes`
- `SELECT table_name, index_name, column_name,  
column_position  
FROM user_ind_columns  
ORDER BY table_name, index_name`

### 3.2.4. Создание последовательностей

*Последовательность* – объект базы данных Oracle, используемый для генерации уникальных порядковых номеров, которые могут использоваться как уникальные идентификаторы записей одной или нескольких таблиц.

Создайте последовательности для генерации уникальных идентификаторов записей таблиц модельной корпоративной базы данных – `s_seq`, `p_seq`, `sp_seq`. Используйте команду PL/SQL `CREATE SEQUENCE`.

Используя документацию по СУБД Oracle, изучите информацию о таблице словаря базы данных `user_sequences`. Протестируйте создание последовательностей с помощью следующего запроса к словарю данных:

```
SELECT sequence_name, min_value, max_value,
 increment_by
FROM user_sequences
```

### 3.2.5. Добавление тестовых данных в таблицы

Добавьте в таблицу `S` записи о следующих поставщиках:

| S# | SName          | SCity        | Address                |
|----|----------------|--------------|------------------------|
| 1  | ЮжУралКомплект | Челябинск    | пр. Ленина, 23, оф. 73 |
| 2  | Технократ      | Москва       | ул. Ямская, 18, оф. 10 |
| 3  | Разноэкспорт   | С.-Петербург | пр. Невский, 45, оф. 3 |
| 4  | Микродеталь    | Магнитогорск | ул. Северная, 4, оф. 1 |
| 5  | АртикулСервис  | Челябинск    | ул. Кирова, 3, оф. 24  |

Для заполнения значения поля **S#** используйте последовательность `s_seq`. Поле **Rating** должно автоматически получить значение по умолчанию в соответствии с заданием. Протестируйте добавление записей с помощью запроса `SELECT * FROM S`.

Добавьте в таблицу `P` записи о следующих деталях:

| P# | PName                | HTP | Weight |
|----|----------------------|-----|--------|
| 1  | Болт 12 мм           | 0   | 20     |
| 2  | Процессор Pentium IV | 1   | 5      |
| 3  | Микросхема RT/3075   | 1   | 15     |
| 4  | Шуруп 11 мм          | 0   | 10     |

| P# | PName                     | HTP | Weight |
|----|---------------------------|-----|--------|
| 5  | Гайка 17 мм               | 0   | 30     |
| 6  | Сейф стальной 90x60x90 см | 0   | 150000 |

Для заполнения значения поля **P#** используйте последовательность `p_seq`. Протестируйте добавление записей с помощью запроса `SELECT * FROM P`.

Добавьте в таблицу `SP` записи о следующих поставках:

| SP# | S# | P# | Qty     | Price   | OrderDate | Period |
|-----|----|----|---------|---------|-----------|--------|
| 1   | 5  | 5  | 7000    | 0.35    | 1-Сен-00  | 10     |
| 2   | 3  | 5  | 1000    | 0.70    | 1-Сен-00  | 5      |
| 3   | 5  | 3  | 4       | 3000.00 | 1-Сен-00  | 5      |
| 4   | 3  | 1  | 5000    | 1.20    | 2-Сен-00  | 2      |
| 5   | 4  | 3  | 10      | 2750.00 | 2-Сен-00  | 8      |
| 6   | 2  | 2  | 100     | 7500.00 | 3-Сен-00  | 10     |
| 7   | 4  | 2  | 300     | 7000.00 | 4-Сен-00  | 7      |
| 8   | 2  | 4  | 1000    | 1.00    | 4-Сен-00  | 4      |
| 9   | 1  | 1  | 2000    | 1.00    | 4-Сен-00  | 1      |
| 10  | 1  | 4  | 2000000 | 0.50    | 5-Сен-00  | 1      |

Для заполнения значения поля **SP#** используйте последовательность `sp_seq`. Поле **ShipDate** добавляемых записей должно иметь значение `NULL`. Протестируйте добавление записей с помощью запроса `SELECT * FROM SP`.

Обратите внимание на то, что в данный момент целостность базы данных нарушена. Во-первых, не изменились рейтинги поставщиков – в соответствии с семантикой предметной области рейтинг поставщика должен увеличиться на 0,1 за каждую поставку. Во-вторых, добавлена поставка (`SP#=10`), вес которой превышает максимально допустимый – в соответствии с семантикой предметной области вес одной поставки не должен превышать 1,5 т. Объясните, почему это произошло.

Выполните запросы SQL, которые приводят данные в согласованное состояние: увеличьте рейтинги поставщиков в соответствии с семантикой предметной области, измените количество деталей в поставке (`SP#=10`) на 10 000, чтобы ее вес не превышал максимально допустимый. Результат выполнения запросов протестируйте.

### 3.2.6. Создание представлений

*Представление* – это виртуальная таблица, представляющая собой настроенное представление данных из одной или нескольких таблиц (называемых базовыми таблицами).

Создайте описанные в 2.5 представления модельной корпоративной базы данных. Используйте команду PL/SQL `CREATE VIEW`. Для реализации полей представления, вычисляемых путем преобразования значения некоторого типа в символьную строку (например, статус поставки словами), используйте встроенную функцию PL/SQL `DECODE` или оператор SQL `CASE`.

Используя документацию по СУБД Oracle, изучите информацию о таблице словаря базы данных `user_views`. Протестируйте создание представлений с помощью следующего запроса к словарю данных:

```
SELECT view_name, text
FROM user_views
```

Проверьте названия и порядок полей, сортировку записей, правильность значений в вычисляемых полях созданных представлений с помощью запросов вида `SELECT * FROM имя_представления`.

Обратите внимание на следующий типичный случай при создании представлений базы данных. Представление `Econ_P` (см. 2.5.2.1) призвано ограничить спектр данных, доступных Экономисту. Для обеспечения удобной работы пользователя это представление содержит вычисляемое поле (`IsНТР` – является ли деталь продуктом высоких технологий словами) и потому является *необновляемым*. Однако в соответствии с требованиями к АРМу Экономиста пользователь должен иметь возможность обновления этого представления (см. 2.5.2.2). Данная коллизия разрешается с помощью создания триггера вида `INSTEAD OF`, который подменяет собой операцию обновления представления `Econ_P`, выполняя обновление соответствующих данных в таблице `P`. Указания по созданию триггеров приведены в 3.2.8.

### 3.2.7. Создание пользователей и наделение их привилегиями

Каждый пользователь базы данных Oracle имеет имя и пароль учетной записи на сервере СУБД Oracle, которые он должен ввести при подключении к базе данных с помощью какого-либо приложения. Пользователь обладает набором *привилегий* – прав выполнения определенных действий над объектами базы данных.

*Роль* – это именованная группа взаимосвязанных привилегий, которую можно назначить пользователям или приложениям базы данных Oracle.

При наделении пользователей привилегиями администратор корпоративной базы данных, как правило, действует следующим образом. Сначала администратор создает роли, отражающие классы пользователей базы данных, и наделяет их соответствующими привилегиями доступа к данным. Затем администратор присваивает пользователям определенного класса соответствующую роль. На последнем шаге, как правило, вместо имени учетной записи пользователя используется его *синоним* (дополнительное имя объекта, ссылка на объект). Синоним имени учетной записи позволяет гибко управлять доступом пользователей к данным в случае кадровых перемещений в организации (поскольку имя учетной записи пользователя на сервере СУБД Oracle, как правило, совпадает с настоящим именем данного пользователя, но не с названием его должности).

Кроме того, администратор корпоративной базы данных ограничивает доступные пользователям системные ресурсы с помощью профилей. *Профиль пользователя* представляет собой именованную совокупность ограничений на потребление системных ресурсов (число одновременных подклю-

чений пользователя к серверу, допустимое время подключения, допустимое время простоя и др.).

☒ Создайте профили пользователей модельной корпоративной базы данных – профиль служащего `emp_profile` и профиль начальника `boss_profile`. Наложите ограничения на потребление системных ресурсов в созданных профилях, как указано в табл. 5.

**Таблица 5. Профили пользователей модельной базы данных**

| Системный ресурс                | Профиль                  |                           |
|---------------------------------|--------------------------|---------------------------|
|                                 | <code>emp_profile</code> | <code>boss_profile</code> |
| Число одновременных подключений | 1                        | 2                         |
| Общее время подключения (часов) | 8                        | 10                        |
| Время простоя (минут)           | 15                       | 30                        |

Используйте команду PL/SQL `CREATE PROFILE`.

Используя документацию по СУБД Oracle, изучите информацию о таблице словаря базы данных `dba_profiles`. Просмотрите информацию о текущих профилях пользователей с помощью запроса `SELECT * FROM dba_profiles`.

☒ Создайте пользователей модельной корпоративной базы данных – `disp_user`, `econ_user`, `boss_user`. Пользователи `disp_user` и `econ_user` должны иметь ограничения на потребление системных ресурсов согласно профилю `emp_profile`, пользователь – согласно профилю `boss_profile`. Используйте команду PL/SQL `CREATE USER`. Наделите созданных пользователей привилегией подключения к базе данных (`CONNECT`). Используйте команду PL/SQL `GRANT`.

Используя документацию по СУБД Oracle, изучите информацию о таблице словаря базы данных `dba_users`. Просмотрите информацию о текущих пользователях базы данных с помощью запроса `SELECT * FROM dba_users`.

Создайте синонимы имен учетных записей, отражающие должность пользователей модельной корпоративной базы данных – `dispatcher` для `disp_user`, `economist` для `econ_user`, `boss` для `boss_user`. Используйте команду PL/SQL `CREATE SYNONYM`.

Используя документацию по СУБД Oracle, изучите информацию о таблице словаря базы данных `user_synonyms`. Протестируйте создание синонимов с помощью запроса `SELECT * FROM user_synonyms`.

☒ Создайте роли пользователей модельной корпоративной базы данных – `disp_role`, `econ_role`, `boss_role`. Используйте команду PL/SQL `CREATE ROLE`.

Используя документацию по СУБД Oracle, изучите информацию о таблице словаря базы данных `dba_roles`. Просмотрите информацию о текущих ролях пользователей с помощью запроса `SELECT * FROM dba_roles`.

☒ Наделите созданные роли привилегиями доступа к данным в соответствии с заданием. Используйте команду PL/SQL GRANT.

☒ Присвойте синонимам имен учетных записей пользователей соответствующие роли. Используйте команду PL/SQL GRANT.

☑ Используя синонимы, наделите пользователей `disp_user`, `econ_user`, `boss_user` привилегиями доступа к данным в соответствии с заданием. Используйте команду PL/SQL GRANT.

Используя документацию по СУБД Oracle, изучите информацию о таблице словаря базы данных `user_tab_privs`. Протестируйте наделение пользователей привилегиями доступа к данным с помощью следующего запроса к словарю данных:

```
SELECT table_name, grantee, privilege, grantable
FROM user_tab_privs
ORDER BY table_name
```

### 3.2.8. Создание триггеров

*Триггер* – особая хранимая процедура, представляющая собой объединение программного кода (тело триггера) и события (событие триггера). Событие триггера – это добавление, удаление или обновление записи таблицы. При наступлении события триггера сервер СУБД Oracle автоматически выполняет тело триггера.

Реализуйте следующие ограничения целостности базы данных и события в базе данных в виде триггеров:

- увеличение рейтинга поставщика на 0,1 при добавлении заказанной у него поставки;
- уменьшение рейтинга поставщика на 0,2 в случае просроченной им поставки;
- запрет поставки высокотехнологичных и дорогих деталей от ненадежных поставщиков;
- ограничение веса одной поставки 1,5 т.

Проанализируйте описание предметной области и определите вид каждого триггера (триггер строки, триггер события, триггер BEFORE/AFTER или какая-либо их комбинация). Используйте команду PL/SQL CREATE TRIGGER.

Реализуйте вставку и обновление записей в необновляемое представление `Econ_P` с помощью триггера вида INSTEAD OF. Используйте команду PL/SQL CREATE TRIGGER. Данный триггер должен использовать последовательность `p_seq` для генерации уникального кода новой детали.

Используя документацию по СУБД Oracle, изучите информацию о таблице словаря базы данных `user_triggers`. Протестируйте создание триггеров с помощью следующего запроса к словарю данных:

```
SELECT table_name, trigger_name, trigger_type,
 triggering_event, when_clause, status
FROM user_triggers
ORDER BY table_name
```



Протестируйте работу триггеров следующим образом:

1. Выполните запрос

```
INSERT INTO SP VALUES (
 sp_seq.nextval,
 1, -- код поставщика "ЮжУралКомплект"
 6, -- код детали "Сейф стальной 90x60x90 см"
 5,
 1000,
 SYSDATE-101,
 1)
```

С помощью запроса

```
SELECT * FROM s
```

убедитесь, что рейтинг поставщика "ЮжУралКомплект" действительно увеличился на 0,1.

2. Выполните запрос

```
UPDATE SP SET
 ShipDate=OrderDate+Period+101
WHERE SP#=11
```

С помощью запроса

```
SELECT * FROM econ_s
```

убедитесь, что рейтинг поставщика "ЮжУралКомплект" уменьшился на 0,6 и данный поставщик теперь является ненадежным.

3. Выполните запрос

```
INSERT INTO SP VALUES (
 sp_seq.nextval,
 1, -- код поставщика "ЮжУралКомплект"
 2, -- код детали "Процессор Pentium IV"
 5000000,
 3800,
 SYSDATE,
 10)
```

Убедитесь, что произошел откат выполнения данного запроса (ненадежный поставщик не может поставлять высокотехнологичную деталь).

4. Выполните запрос

```
INSERT INTO SP VALUES (
 sp_seq.nextval,
 1, -- код поставщика "ЮжУралКомплект"
 6, -- код детали "Сейф стальной 90x60x90 см"
 11,
 1001,
 SYSDATE,
 10)
```

Убедитесь, что произошел откат выполнения данного запроса (ненадежный поставщик не может поставлять дорогую деталь).

## 5. Выполните запрос

```
INSERT INTO SP VALUES (
 sp_seq.nextval,
 1, -- код поставщика "ЮжУралКомплект"
 6, -- код детали "Сейф стальной 90x60x90 см"
 11,
 999,
 SYSDATE,
 10)
```

Убедитесь, что произошел откат выполнения данного запроса (вес добавляемой поставки превышает максимально допустимый).

### 3.2.9. Создание пакета хранимых процедур

*Хранимая процедура* или *функция* – это подпрограмма на языке PL/SQL, хранящаяся в базе данных Oracle. Процедуры и функции и могут быть объединены в *пакет*.

Создайте пакет, содержащий следующие подпрограммы<sup>3</sup>:

- `calc_delay(SP# IN number, delay OUT number)`  
вычисление превышения срока поставки в днях для поставки с заданным кодом;
- `calc_penalty_for_sp(SP# IN number, penalty OUT number)`  
вычисление суммы неустойки за просроченную поставку для поставки с заданным кодом;
- `calc_exceeding_qty(s IN number, start IN date, stop IN date, qty OUT number)`  
вычисление количества просроченных поставок у поставщика с заданным кодом в заданный период;
- `calc_penalty_of_s(s IN number, start IN date, stop IN date, penalty OUT number)`  
вычисление общей суммы неустоек, предъявленных поставщику с заданным кодом в заданный период.

Используйте команды PL/SQL `CREATE PACKAGE`, `CREATE PROCEDURE`, `CREATE FUNCTION`, `CREATE PACKAGE BODY`.

Используя документацию по СУБД Oracle, изучите информацию о таблице словаря базы данных `user_source`. Протестируйте создание подпрограмм с помощью следующего запроса к словарю данных:

```
SELECT name, type,
FROM user_source
ORDER BY type, name
```

Протестируйте выполнение подпрограмм, используя тесты в табл. 6 (при заданных входных данных результат должен совпадать с ожидаемым).

---

<sup>3</sup> Подпрограммы специфицированы как процедуры. Вы можете оформить их в виде функций.

**Таблица 6. Данные для тестирования хранимых процедур**

| Подпрограмма        | Входные данные                                      | Ожидаемый результат |
|---------------------|-----------------------------------------------------|---------------------|
| calc_delay          | SP# => 11                                           | 100                 |
| calc_penalty_for_sp | SP# => 11                                           | 500.00              |
| calc_exceeding_qty  | s => 1,<br>start => SYSDATE-1,<br>stop => SYSDATE+1 | 1                   |
| calc_penalty_of_s   | s => 1,<br>start => 1-Дек-00,<br>stop => 31-Дек-00  | 120000.00           |

Наделите соответствующие роли привилегией исполнения (EXECUTE) созданных подпрограмм. Используйте команду PL/SQL GRANT.

Используя синонимы, наделите соответствующих пользователей disp\_user, econ\_user, boss\_user привилегией исполнения (EXECUTE) созданных подпрограмм. Используйте команду PL/SQL GRANT.

### 3.3. Разработка АРМов

При выполнении указаний данного раздела Вы выступаете в роли прикладного программиста, создающего приложение для базы данных Oracle. Вам предстоит разработать клиентские приложения модельной корпоративной базы данных Oracle с использованием СУБД MS Access.

Перед выполнением указаний данного раздела подключитесь к базе данных Oracle как пользователь admin\_user и удалите все записи из таблиц S, P, SP.

При установлении связи приложения MS Access с таблицами и другими объектами базы данных Oracle необходимо подключаться к базе данных Oracle в качестве пользователя соответствующего АРМа – disp\_user, econ\_user, или boss\_user.

С помощью Мастера ответов встроенной справки MS Access изучите темы "Импорт и связывание данных", "Запрос к серверу", "Строка подключения ODBC".

Далее приведены указания по реализации функций, назначенных кнопкам и/или другим элементам управления диалоговых форм АРМов.

#### 3.3.1. АРМ диспетчера

Создайте базу данных Диспетчер.mdb и добавьте в нее связанные таблицы disp\_s, disp\_sp и disp\_shipment модельной корпоративной базы данных Oracle. Переименуйте связанные таблицы на Поставщики, Поставки и Доставка соответственно.

Разработайте диалоговые формы, указанные в спецификации АРМа (см. 2.5.1.4). С помощью форм добавьте в базу данных тестовые данные о поставщиках, указанные в 3.2.5.

Обратите внимание на следующее обстоятельство. Форма "Поставщики" (см. 2.5.1.4) содержит кнопку "Добавить поставщика...", нажатие которой предполагает вставку записи в (обновляемое) представление `Disp_S`. При этом значение поля `S#` новой записи должно быть получено с помощью последовательности `s_seq`. СУБД MS Access, однако, не предоставляет удобных средств для реализации данной функции АРМа. Если обработка события нажатия указанной кнопки реализуется с помощью запроса к серверу СУБД Oracle, то такой запрос не может содержать ссылки на данные из формы MS Access. Если же обработка события реализуется с помощью запроса СУБД MS Access, то такой запрос не может содержать ссылки на объекты базы данных Oracle (то есть ссылку на последовательность `s_seq`). Данная коллизия разрешается следующим образом. Обработку нажатия кнопки следует реализовать как запрос MS Access на добавление записи, который использует данные из соответствующей формы (не содержащей нового значения поля `S#`). Вместе с тем следует написать триггер вида `INSTEAD OF`, который подменит собой вставку записи в представление `Disp_S`, используя при этом последовательность `s_seq` для генерации уникального кода нового поставщика.

Используя тестовые данные из 3.2.8, с помощью диалоговых форм протестируйте работу триггеров базы данных.

Для реализации отчетов, указанных в спецификации АРМа (см. 2.5.1.3) создайте в схеме пользователя `admin_user` представления "Данные для оформления путевых листов", "Данные для оформления требований об оплате неустойки за просроченные поставки", "Список просроченных поставок за заданный период". Используйте ранее написанные подпрограммы `calc_delay` и `calc_penalty_for_sp`. Наделите пользователя `disp_user` привилегией выборки данных из этих представлений. Реализуйте и протестируйте отчеты.

### **3.3.2. АРМ экономиста**

Создайте базу данных `Экономист.mdb` и добавьте в нее связанные таблицы `econ_s`, `econ_p` и `econ_sp` модельной корпоративной базы данных Oracle. Переименуйте связанные таблицы на Поставщики, Детали и Поставки соответственно.

Разработайте диалоговые формы, указанные в спецификации АРМа (см. 2.5.2.4). С помощью форм добавьте в базу данных тестовые данные о деталях и поставках, указанные в 3.2.5.

Используя тестовые данные из 3.2.8, с помощью диалоговых форм протестируйте работу триггеров базы данных.

Для реализации добавления записи в таблицу Детали создайте соответствующий запрос на добавление. СУБД Oracle автоматически подменит выполнение данного запроса выполнением триггера `INSTEAD OF`, который обсуждался в 3.2.8.

Для реализации отчетов, указанных в спецификации АРМа (см. 2.5.2.3) создайте в схеме пользователя `admin_user` представления "Надежные поставщики", "Ненадежные поставщики", "Дорогие детали", "Дешевые детали", "Высокотехнологичные детали", "Низкотехнологичные детали". Наделите пользователя `econ_user` привилегией выборки данных из этих представлений. Реализуйте и протестируйте отчеты.

### 3.3.3. АРМ начальника отдела

Создайте базу данных `Начальник.mdb` и добавьте в нее связанные таблицы `boss_s`, `boss_p` и `boss_sp` модельной корпоративной базы данных Oracle. Переименуйте связанные таблицы на Поставщики, Детали и Поставки соответственно.

Разработайте диалоговые формы, указанные в спецификации АРМа (см. 2.5.3.4).

Для реализации отчетов, указанных в спецификации АРМа (см. 2.5.3.3) создайте в схеме пользователя `admin_user` представления "Список поставщиков с указанием суммы их поставок", "Список деталей с указанием суммы их поставок", "Список поставщиков с указанием количества просроченных ими поставок и общей суммы предъявленных им неустоек". Используйте ранее написанные подпрограммы `calc_exceeding_qty` и `calc_penalty_of_s`. Наделите пользователя `boss_user` привилегией выборки данных из этих представлений. Реализуйте и протестируйте отчеты.

## Список рекомендуемой литературы

1. *Баженова И.Ю.* Oracle 8: Уроки программирования. М.: Диалог-МИФИ, 2000.
2. *Дорси П., Колетски П.* Oracle Designer: Настольная книга пользователя. М.: ЛОРИ, 2000.
3. *Задворьев И.С., Смирнов С.Н.* Работаем с Oracle. М.: Гелиос, 2000.
4. *Луни К.* Oracle 8i: Настольная книга администратора. М.: ЛОРИ, 2000.
5. *Тимошок Т.В.* Microsoft Access 2002: Самоучитель. М.: Диалектика, 2003.
6. *Урманн С.* Oracle 8: Программирование на языке PL/SQL. М.: ЛОРИ, 1999.
7. *Эбби М., Кори М.* Oracle 8: Первое знакомство. М.: ЛОРИ, 1998.