



# ТЕМПОРАЛЬНЫЕ БАЗЫ ДАННЫХ

Время – это деньги, за которые мы должны купить вечность.

*Х. де Балангер*

Пост-реляционные системы баз данных

# Содержание

- Понятие темпоральных данных и темпоральных СУБД
- Линии и гранулярность времени
- Интервальное и точечное представления времени
- Архитектура темпоральной СУБД
- Язык запросов к темпоральным данным
- Примеры темпоральных СУБД

# Темпоральные данные и СУБД

3

- *Темпоральные данные* – это произвольные данные, которые явно или неявно связаны с определенными датами или промежутками времени.
- *Темпоральные СУБД* – это СУБД для хранения и обработки темпоральных баз данных.
  - Темпоральными не являются обычные реляционные СУБД
    - Связанные со временем типы данных используются для хранения данных о *текущем* состоянии.
    - Интерпретацией и связью данных (или событий) между собой с учетом времени занимается разработчик.
  - В темпоральной СУБД с каждым данным связывается *период истинности* этого данного.

# Действительное время

4

- *Действительное (модельное) время* показывает период в прошлом, настоящем или будущем, когда факт являлся истинным в моделируемом мире, представленном в базе данных.

| ФИО    | Зарплата |
|--------|----------|
| Иванов | 500      |
| Петров | 400      |

| ФИО    | Зарплата | Действительное время |          |
|--------|----------|----------------------|----------|
|        |          | Начало               | Конец    |
| Иванов | 500      | 01.01.09             | СЕЙЧАС   |
| Петров | 300      | 01.03.08             | 31.01.08 |
| Петров | 400      | 01.02.09             | СЕЙЧАС   |

# Транзакционное время

5

- *Транзакционное время* показывает период в прошлом или настоящем, когда данная запись была представлена в базе данных.
- Операция обновления записи понимается как составная операция удаления старой записи и добавления новой.

| ФИО    | Зар-плата | Действительное время |          |
|--------|-----------|----------------------|----------|
|        |           | Начало               | Конец    |
| Иванов | 500       | 01.01.09             | СЕЙЧАС   |
| Петров | 300       | 01.03.08             | 31.01.09 |
| Петров | 400       | 01.02.09             | СЕЙЧАС   |

| ФИО    | Зар-плата | Транзакционное время |          |
|--------|-----------|----------------------|----------|
|        |           | Начало               | Конец    |
| Иванов | 500       | 20.12.08             | СЕЙЧАС   |
| Петров | 300       | 03.03.08             | 27.01.09 |
| Петров | 500       | 28.01.09             | 05.02.09 |
| Петров | 400       | 06.02.09             | СЕЙЧАС   |

# Две линии времени

6

- Метки действительного времени хранят информацию об изменении некоторых параметров моделируемого мира.
- Метки транзакционного времени предоставляют информацию о времени изменения данных или исправления ошибок.

| ФИО    | Зарплата | Действительное время |          | Транзакционное время |          |
|--------|----------|----------------------|----------|----------------------|----------|
|        |          | Начало               | Конец    | Начало               | Конец    |
| Иванов | 500      | 01.01.09             | СЕЙЧАС   | 20.12.08             | СЕЙЧАС   |
| Петров | 300      | 01.03.08             | 31.01.09 | 03.03.08             | 27.01.09 |
| Петров | 500      | 01.02.09             | СЕЙЧАС   | 28.01.09             | 05.02.09 |
| Петров | 400      | 01.02.09             | СЕЙЧАС   | 06.02.09             | СЕЙЧАС   |

# Гранулярность времени

7

- *Гранулярность* показывает, насколько близкие моменты на оси времени все еще будут отличимыми друг от друга.
  - Для данных о заработной плате сотрудника достаточно разбиения по дням.
  - Для транзакционного времени может быть недостаточно разбиения по секундам, если в СУБД возможна более частая фиксация транзакций.

# Интервальное и точечное представление

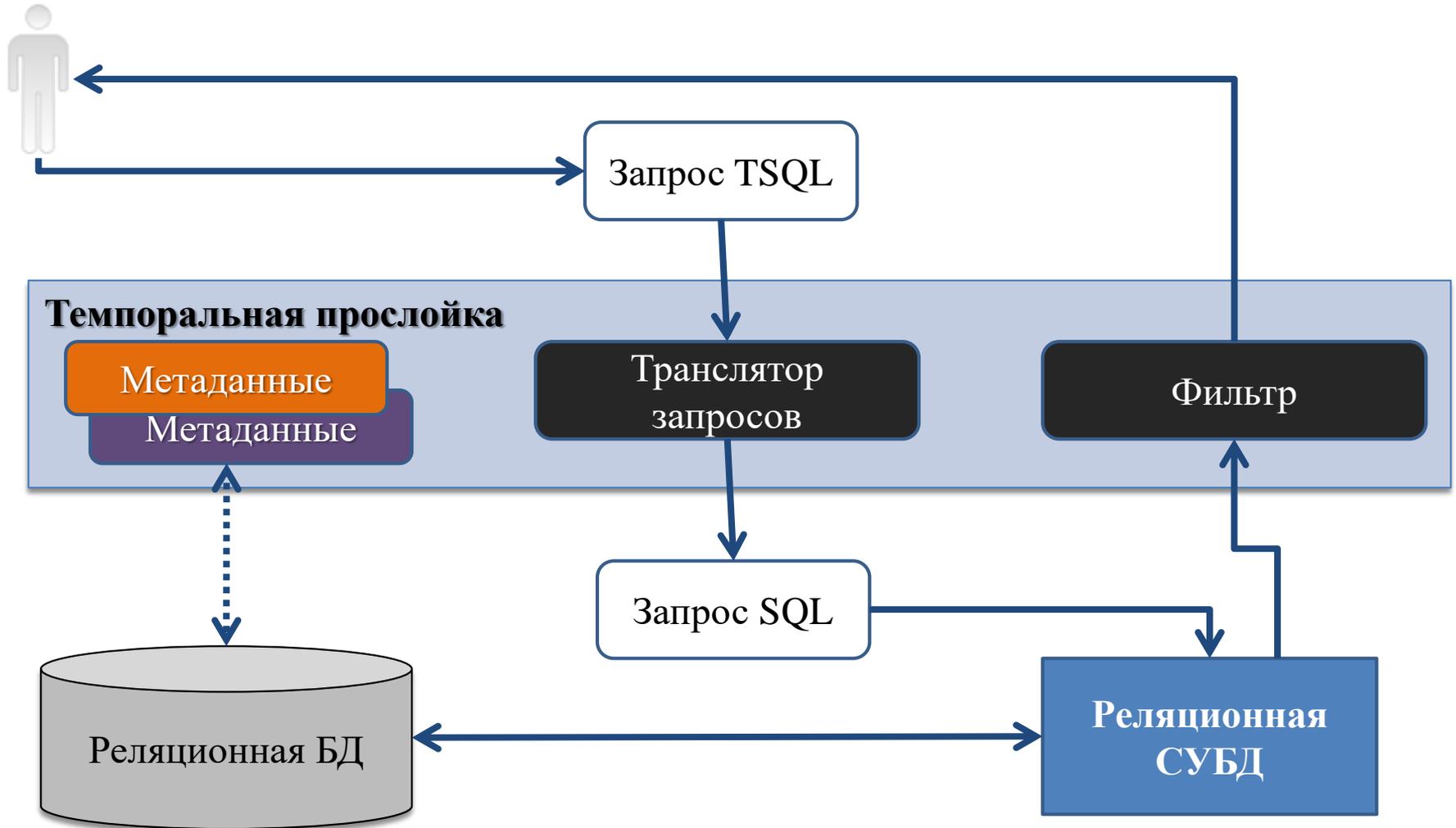
8

- *Интервальное* представление темпоральных данных связывает с фактом временной интервал истинности данного факта.
- *Точечное* представление темпоральных данных связывает отдельный момент времени и все факты, которые были истинны в данный момент.

| Дата     | Действит. время |        | Транзакц. время |        |
|----------|-----------------|--------|-----------------|--------|
|          | Иванов          | Петров | Иванов          | Петров |
| 01.03.08 | NULL            | 300    | NULL            | NULL   |
| 03.03.08 | NULL            | 300    | NULL            | 300    |
| 20.12.08 | NULL            | 300    | 500             | 300    |
| 01.01.09 | 500             | 300    | 500             | 300    |
| 27.01.09 | 500             | 300    | 500             | 300    |
| 28.01.09 | 500             | 300    | 500             | 500    |
| 31.01.09 | 500             | 300    | 500             | 500    |
| 01.02.09 | 500             | 400    | 500             | 500    |
| 05.02.09 | 500             | 400    | 500             | 500    |
| 06.02.09 | 500             | 400    | 500             | 400    |
| ...      | 500             | 400    | 500             | 400    |
| СЕЙЧАС   | 500             | 400    | 500             | 400    |

# Разработка темпоральной СУБД

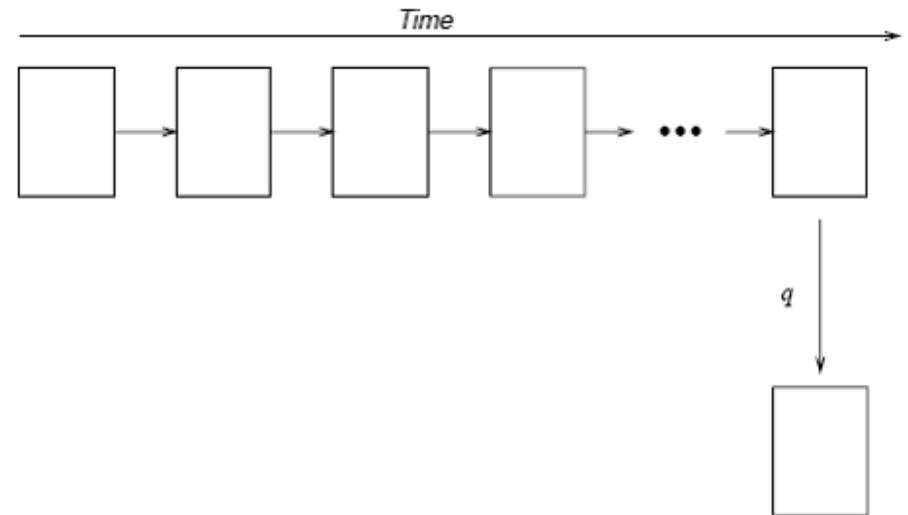
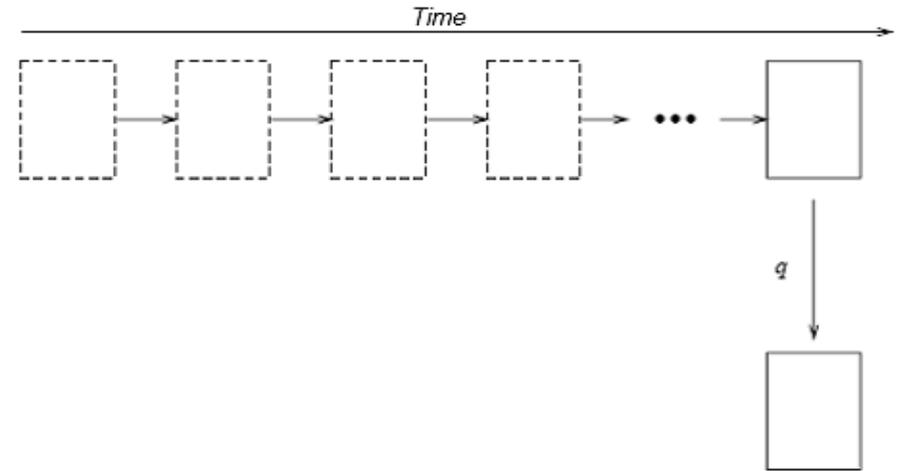
9



# Поддержка темпоральных запросов

10

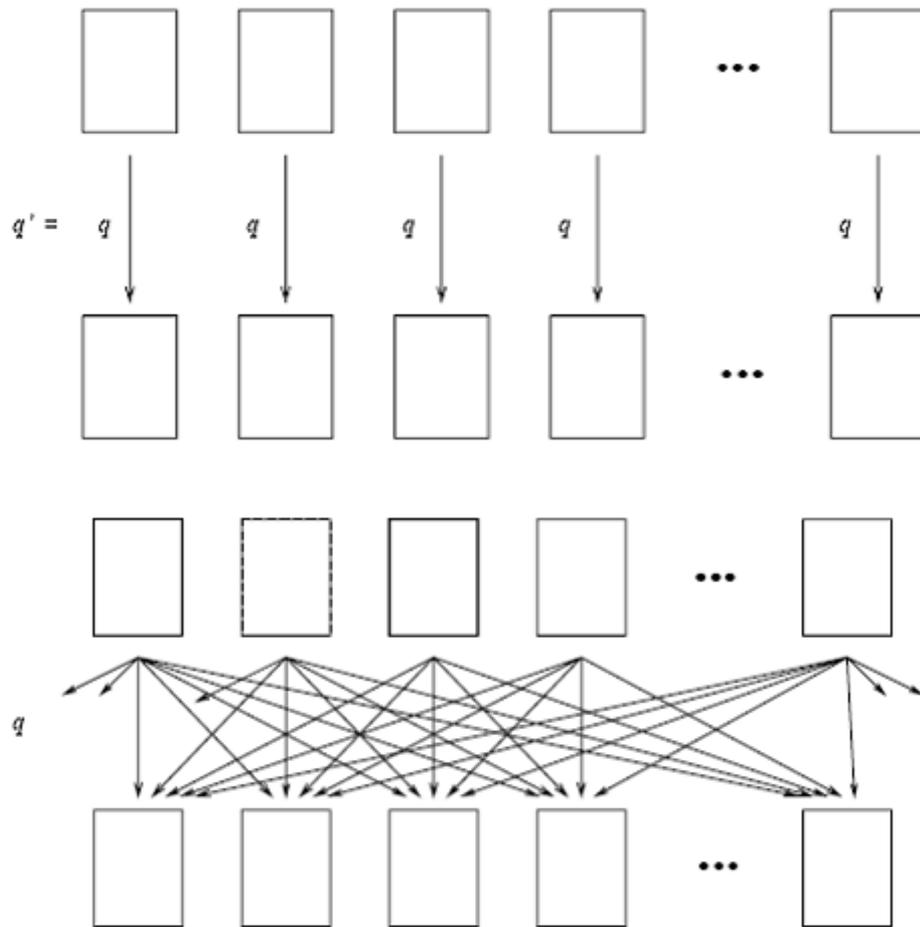
- Нет поддержки
  - ▣ Выдать текущую зарплату сотрудников.
- Реляционный запрос к таблице с темпоральной поддержкой
  - ▣ Выдать зарплату, полученную каждым сотрудником в указанную дату.



# Поддержка темпоральных запросов

11

- **Последовательные темпоральные запросы**
  - Выдать периоды и соответствующие размеры зарплаты каждого сотрудника.
  
- **Произвольные темпоральные запросы**
  - Выдать периоды, когда на оплату труда сотрудников уходила максимальная сумма.



# Язык темпоральных запросов

12

## □ Создание таблицы

```
□ CREATE TABLE Emp (  
    EmpID INTEGER,  
    Name CHAR(30),  
    Dept CHAR(40),  
    Salary INTEGER)  
AS VALIDTIME AND TRANSACTIONTIME;
```

## □ Вставка данных

```
□ VALIDTIME PERIOD '03.01.08-31.01.09'  
INSERT INTO Emp VALUES (  
    10, 'Петров', 'Отдел поставок', 300);  
  
□ VALIDTIME PERIOD '01.02.08-NOW'  
INSERT INTO Emp VALUES (  
    10, 'Петров', 'Отдел поставок', 500);
```

# Язык темпоральных запросов

13

## □ Выборка данных

### ▣ VALIDTIME

```
SELECT * FROM Emp;
```

### ▣ TRANSACTIONTIME

```
SELECT * FROM Emp;
```

### ▣ VALIDTIME AND TRANSACTIONTIME

```
SELECT * FROM Emp;
```

## □ Темпоральные ограничения целостности

### ▣ CREATE TABLE Emp (

```
EmpID INTEGER,
```

```
Name CHAR(30),
```

```
Dept CHAR(40) VALIDTIME REFERENCES Depts(DeptID),
```

```
Salary INTEGER)
```

```
AS VALIDTIME AND TRANSACTIONTIME;
```

# Представление результатов темпоральных запросов

14

- Для хранения данных в базе данных наиболее компактным является интервальное представление. При формулировке запросов наиболее удобным является точечное представление (удобнее сравнивать моменты времени, а не оперировать интервалами истинности отдельных фактов).
- Интервалы времени принято разделять на такие подинтервалы, чтобы для каждого факта они не пересекались, но дополняли друг друга до исходного интервала, а любые два интервала разных фактов либо не пересекались, либо совпадали – операция *распаковки*.
  - При таком представлении истинность фактов не будет меняться ни для одного из получившихся интервалов, что позволит корректно использовать полный вложенный перебор по интервалам и фактам при получении результата запросов.
- После получения результата в виде набора интервалов истинности для каждого факта необходимо выполнить обратную процедуру *упаковки* – объединение пересекающихся или примыкающих друг к другу интервалов.

# Значение "СЕЙЧАС"

15

- Специальное значение *СЕЙЧАС* используется при задании верхней границы интервала истинности факта, хранимого в таблице с темпоральной поддержкой.
  - ▣ *СЕЙЧАС*  $\neq$  CURRENT\_TIMESTAMP, CURRENT\_DATE или CURRENT\_TIME.
  - ▣ При выборе значения для представления *СЕЙЧАС* необходимо гарантировать, чтобы это значение не использовалось с каким-либо иным смыслом.
  - ▣ В качестве *СЕЙЧАС* могут быть взяты значение NULL либо максимальное или минимальное значение типа TIMESTAMP.
- При использовании *СЕЙЧАС* в запросах необходимо принимать во внимание, что его следует заменять текущим временем на этапе выполнения. Эксперименты с СУБД TimeDB показали, что наиболее эффективно выполнение запросов при *СЕЙЧАС*=max(TIMESTAMP).

# Разрежение темпоральных данных

16

- Постоянно возрастающий объем хранимых данных – проблема систем с темпоральной поддержкой транзакционного времени.
- *Разрежение (vacuuming)* – физическое удаление устаревших данных (например, удаленных до определенного момента времени).
- Разрежение повышает производительность, но может привести к некорректной работе темпоральных запросов (например, использующих агрегатные функции).

# Проектирование темпоральных баз данных

17

- Сколько в базе данных должно быть таблиц с темпоральной поддержкой?
  - Действительное время: исходя из потребностей задачи.
  - Транзакционное время: исходя из возможностей СУБД.
- Дополнение обычных реляционных таблиц колонками для отметки действительного времени.
  - Все "обычные" реляционные ключи таблиц будут неявно расширяться верхней границей интервала действительного времени. Ограничения целостности также должны формулироваться с учетом этих обстоятельств .
- Как обрабатывать запросы на выборку одновременно из таблицы с поддержкой транзакционного времени и из таблицы без подобной поддержки?
  - Разрешить такие запросы: результат выборки будет неверен, так как информация из обычной таблицы могла быть уже удалена или изменена, а в соответствующей темпоральной таблице остались внешние связи.
  - Запретить такие запросы: как работать с потенциально константными данными (например, названия дней недели), которые могут храниться в таблице без темпоральной поддержки.
  - Принудительное добавление темпоральной поддержки для всех таблиц: не лучший выход, так как усложняется алгоритм работы СУБД и требуются дополнительные ресурсы.

# ACID-свойства темпоральных транзакций

18

- ACID-свойства темпоральных SQL-транзакций могут быть сохранены за счет отображения каждой темпоральной транзакции в одиночную реляционную SQL транзакцию.
- Для каждой транзакции необходимо обеспечить, чтобы в ее пределах транзакционное время было константно.
  - Транзакционное время выбирается *в конце выполнения транзакции*, после того, как получены блокировки на все затрагиваемые объекты
  - Если транзакционное время выбирается в начале выполнения транзакции, то параллельная транзакция может внести изменения в таблицы, которые еще только понадобятся данной транзакции.

# Смежные области исследований

19

- Темпоральные базы XML-данных и темпоральное расширение языка XQuery.
- Базы данных мультимедиа.
- Пространственно-временные базы данных (о перемещения объектов, получаемые с GPS-датчиков).
- Снимки состояний базы данных.
- Версионирование темпоральных данных.

# Темпоральные СУБД

20

- TimeDB
  - Свободно распространяемый прототип темпоральной СУБД, <http://www.timeconsult.com/Software/Software.html>
  - Реализован на Java, JDBC
- TimeSeries Datablade (Informix)
  - Предназначен для обработки и анализа динамики процессов на основе временных рядов данных.
  - Содержит определение новых типов данных – временного ряда и календаря, а также предоставляет функций для обработки данных, содержащих временные метки.

# Темпоральные СУБД

21

- Immortal DB (Microsoft)
  - Поддержка транзакционного времени встроена в SQL-сервер.
- Технология Flashback (Oracle)
  - Позволяет пользователям без участия администратора и внесения каких-либо структурных изменений в базу данных просмотреть состояние базы данных на какой-либо момент в прошлом и исправить пользовательские ошибки.
  - Для использования Flashback Query требуется включение автоматического управления восстановлением (Automatic Undo Management) вместо использования сегментов отката (Rollback Segments).