ВЫБОРКА ДАННЫХ ИЗ ОДНОЙ И НЕСКОЛЬКИХ ТАБЛИЦ. АГРЕГАТНЫЕ ФУНКЦИИ В SQL

Лектор: PhD, и.о. доцента Турарбек Әсем Турарбекқызы

Цель лекции

Изучить методы выборки данных из одной и нескольких таблиц с использованием SELECT, соединений JOIN, агрегатных функций и вложенных запросов.

Результаты обучения

По завершении лекции студент должен:

1. **Формировать запросы SELECT с использованием DISTINCT и WHERE.**
2. **Объединять данные нескольких таблиц с помощью INNER JOIN, LEFT JOIN, RIGHT JOIN и FULL JOIN.**
3. **Применять агрегатные функции, GROUP BY, HAVING и вложенные запросы.**

Выборка данных из одной таблицы

Основы оператора SELECT

Команда SELECT является центральной в SQL и реализует проекцию и выборку — две основные операции реляционной алгебры.

- Проекция (π) — выбор нужных столбцов.
- Выборка (σ) — фильтрация строк по условию.

Общая структура запроса:

```
SELECT [DISTINCT] столбцы
FROM имя_таблицы
[WHERE условие]
[GROUP BY столбец]
[HAVING агрегатное_условие]
[ORDER BY столбец [ASC|DESC]]
[LIMIT число];
```

- DISTINCT устраняет дубликаты.
- WHERE фильтрует строки до агрегации.
- ORDER BY выполняется последним, и не влияет на группы или фильтрацию.

ВЫБОРКА ИЗ НЕСКОЛЬКИХ ТАБЛИЦ

Общая структура запроса:

Реляционные базы данных проектируются в нормализованном виде, поэтому сущности часто распределены по нескольким таблицам. Для извлечения связанной информации применяется операция соединения (**JOIN**).

Теория соединений

Согласно реляционной алгебре (C. J. Date), операция соединения — это форма декартова произведения с последующим отбором нужных строк.

Внешний ключ обеспечивает логическую связь между таблицами: он указывает на первичный ключ другой таблицы.

JOIN	Формальное имя (алгебра)	Описание
INNER JOIN	θ-соединение (theta join)	Строки обеих таблиц, удовлетворяющие условию соединения
LEFT JOIN	внешнее левое соединение	Все строки левой таблицы + совпадения из правой
RIGHT JOIN	внешнее правое соединение	Все строки правой таблицы + совпадения из левой
FULL JOIN	полное внешнее соединение	Все строки обеих таблиц
JOIN	**Формальное имя (алгебра)**	**Описание**

Примеры JOIN

Пример (INNER JOIN):

SELECT С.ФИО, О.Название

FROM Сотрудники С

JOIN Отделы О ON С.Отдел_ID = О.Отдел_ID;

Здесь Сотрудники.Отдел_ID — **внешний ключ**, а Отделы.Отдел_ID — **первичный ключ**.

Множественная сортировка: сначала по отделу, затем по фамилии:

SELECT Отдел, ФИО

FROM Сотрудники

ORDER BY Отдел ASC, ФИО ASC;

FULL OUTER JOIN

Таблица: Сотрудники

id	имя	отдел_id
1	Айгүл	10
2	Нурбол	20
3	Алихан	30

Таблица: Отделы

id	название
10	IT
20	Финансы
40	Маркетинг

```
SELECT имя, название
FROM Сотрудники
FULL OUTER JOIN Отделы
ON Сотрудники.отдел_id = Отделы.id;
```

Возвращает таблицу: все строки из обеих таблиц

имя	название
Айгүл	IT
Нурбол	Финансы
Алихан	NULL
NULL	Маркетинг

АГРЕГАТНЫЕ ФУНКЦИИ В SQL

Основные агрегатные функции:

Агрегатные функции реализуют групповую обработку данных.

Они позволяют анализировать множественные строки как единую статистическую группу. Эти функции соответствуют агрегатам в реляционной алгебре.

```
SELECT Клиент_ID, SUM(Сумма)
FROM Заказы
GROUP BY Клиент_ID;
```

Группировка (GROUP BY) используется для агрегации по значениям одного или нескольких столбцов.

Функция	Описание
COUNT()	Подсчитывает количество строк или непустых значений
SUM()	Считает сумму значений числового столбца
AVG()	Вычисляет среднее значение числового столбца
MAX()	Возвращает наибольшее значение
MIN()	Возвращает наименьшее значение

Примеры

Условия после агрегации — HAVING

Оператор HAVING используется для фильтрации агрегированных групп, тогда как WHERE фильтрует индивидуальные строки до группировки.

```
SELECT Отдел_ID, AVG(Зарплата)

FROM Сотрудники

GROUP BY Отдел_ID

HAVING AVG(Зарплата) > 200000;
```

Вложенные запросы и агрегаты

Агрегаты часто используются во вложенных подзапросах:

```
SELECT ФИО

FROM Сотрудники

WHERE Зарплата > (

    SELECT AVG(Зарплата) FROM Сотрудники

);
```

Возвращает сотрудников с зарплатой выше среднего.

Семантика и порядок выполнения SELECT-запроса

Реальный **порядок выполнения** SQL-запроса отличается от его синтаксического порядка:

1. **FROM + JOIN**
2. **WHERE**
3. **GROUP BY**
4. **HAVING**
5. **SELECT**
6. **ORDER BY**
7. **LIMIT**

Заключение

SELECT является основным инструментом извлечения данных в SQL. Соединения JOIN позволяют получать взаимосвязанную информацию из нескольких таблиц, а агрегатные функции и вложенные запросы расширяют возможности анализа данных.

Вопросы для самопроверки

1. Какова общая структура запроса SELECT?
2. Для чего используется DISTINCT?
3. Чем INNER JOIN отличается от LEFT JOIN и FULL JOIN?
4. Как применяются агрегатные функции с GROUP BY и HAVING?
5. Для чего используются вложенные SQL-запросы?

Список использованных источников

1. Silberschatz A., Korth H. F., Sudarshan S. *Database System Concepts*. 7th ed. McGraw-Hill, 2020.
2. Elmasri R., Navathe S. B. *Fundamentals of Database Systems*. 7th ed. Pearson, 2016.
3. Beaulieu A. *Learning SQL*. 3rd ed. O'Reilly Media, 2020.
4. Molinaro A., de Graaf R. *SQL Cookbook*. 2nd ed. O'Reilly Media, 2020.
5. Garcia-Molina H., Ullman J. D., Widom J. *Database Systems: The Complete Book*. 2nd ed. Pearson, 2009.
6. PostgreSQL Global Development Group. *PostgreSQL Documentation: SELECT*. https://www.postgresql.org/docs/current/sql-select.html
7. PostgreSQL Global Development Group. *PostgreSQL Documentation: Table Expressions*. https://www.postgresql.org/docs/current/queries-table-expressions.html
8. Oracle Corporation. *MySQL 8.4 Reference Manual: JOIN Clause*. https://dev.mysql.com/doc/refman/8.4/en/join.html

СПАСИБО ЗА ВНИМАНИЕ