



ПОЛИТЕХ

Санкт-Петербургский
политехнический университет
Петра Великого

Институт дополнительного образования
Высшая инженерная школа

DEV-J130. Java SE. Разработка многоуровневых приложений.
Введение в язык SQL.



Рассматриваемые вопросы

- Основные термины и понятия.
- Structured Query Language (SQL) - язык структурированных запросов.



Общие термины

- ▣ **База данных** — совокупность данных, организованных в соответствии с концептуальной структурой, описывающей характеристики этих данных и взаимоотношения между ними, которая поддерживает одну или более областей применения (ISO/IEC 2382:2015 - Information technology. Vocabulary).
- ▣ **База данных** — совокупность данных, хранимых в соответствии с схемой данных, манипулирование которыми выполняют в соответствии с правилами средств моделирования данных (ГОСТ 33707—2016. Информационные технологии. Словарь).



Общие термины

- ▣ **Система управления базами данных (СУБД, DBSM)** – совокупность программных и лингвистических средств общего или специального назначения, обеспечивающих управление созданием и использованием баз данных (ГОСТ 33707—2016. Информационные технологии. Словарь).



Общие термины

- **SQL** (Structured Query Language) — декларативный язык программирования, применяемый для создания, модификации и управления данными в реляционной базе данных, управляемой соответствующей системой управления базами данных.
- Язык SQL стандартизован Международной организацией по стандартизации (ISO – International Organization for Standardization) и Американским национальным институтом стандартов (ANSI - American National Standards Institute) и ISO.



«Диалекты» SQL

- В стандарте SQL-92 было введено понятие «уровня соответствия».
- ANSI и NIST определяют четыре уровня соответствия реализации стандарту SQL:
 - Entry (базовый);
 - Transitional (переходный);
 - Intermediate (промежуточный);
 - Full (полный).



Основные команды языка SQL



Комментарии SQL

- В SQL можно использовать два стандартных вида комментариев:
 - -- Однострочный комментарий

 - /*
 - * Многострочный
 - * комментарий
 - */



Встроенные типы языка SQL

- ❑ Система встроенных типов языка SQL не совпадает с системой встроенных типов Java.
- ❑ Некоторые часто используемые типы SQL:
 - ❑ INTEGER - целочисленный тип, совпадающий с типом int в Java;
 - ❑ DOUBLE (size, fract) - вещественный тип;
 - ❑ CHAR(size) - символьный тип длиной до 255 символов;
 - ❑ VARCHAR (size) - символьный тип (строка) переменной длины;
 - ❑ TEXT – символьный тип (текст) длиной до 65 535 символов;
 - ❑ DATETIME – дата со временем;
 - ❑ DATE – дата.



Встроенные константы

- NULL - встроенная константа, которая означает неопределённое значение, которая совместима со всеми встроенными типами данных.
- В качестве значений логического типа `BOOL(EAN)` используются встроенные константы `TRUE`, `FALSE` и `NULL`.



Основные ограничения

- PRIMARY KEY — первичный ключ (идентификатор) записи в текущей таблице;
- FOREIGN KEY — внешний ключ (идентификатор) записи;
- NOT NULL — запрет на неопределённое значение;
- DEFAULT value — определение значения по умолчанию;
- UNIQUE — требование уникальности значения поля;
- CHECK — задание обязательного условия для значений поля;
- INDEX — индексация по данному полю.



Подмножество DDL

- **Язык определения данных** (DDL — Data Definition Language) – подмножество языка SQL, команды которого предназначены для определения структурной модели данных.
- Основные команды DDL:
 - **CREATE** — определяет отношение (таблицу), представление или другой объект базы данных;
 - **ALTER** — модифицирует существующий в базе данных объект, например, таблицу;
 - **DROP** — удаляет существующий объект из базы данных.



Команда CREATE TABLE

- Упрощённый синтаксис:
 - **CREATE TABLE** table-name (field-name type constraints, ... , CONSTRAINT constraint-name constraint-definition ...);
- Примеры:
 1. `CREATE TABLE a (id INTEGER PRIMARY KEY, title VARCHAR(16) NOT NULL);`
 2. `CREATE TABLE b (id INTEGER PRIMARY KEY, id_a INTEGER, comment VARCHAR(255), CONSTRAINT fk_a FOREIGN KEY(id_a) REFERENCES a(id));`



Команда ALTER TABLE

- Упрощённый синтаксис:

```
ALTER TABLE table-name
```

```
        ADD | ALTER/MODIFY | DROP COLUMN column-name  
data-type;
```

- Пример:

```
1. ALTER TABLE a ADD COLUMN notes VARCHAR(255);
```



Команда DROP TABLE

- Упрощённый синтаксис:
 - **DROP TABLE** table-name;
- Пример:
 - **DROP TABLE** a;



Подмножество DML

- **Data Manipulation Language (DML)** — подмножество языка SQL, команды которого предназначены для изменения и чтения данных.
- Основные команды:
 - **INSERT** — команда добавления новой записи в базу данных;
 - **UPDATE** — команда изменения существующей записи;
 - **DELETE** — команда удаления записи из базы данных;
 - **SELECT** — команда чтения записей из базы данных.



Команда INSERT INTO

- Упрощённый синтаксис:

- **INSERT INTO** table (field-list) VALUES (value-list);

- Пример:

- 1. **INSERT INTO a (id, title) VALUES (1, 'first record');**

- 2. **INSERT INTO a VALUES (2, 'second record');**



Выражение WHERE

- Выражение WHERE используется как часть команды для определения условия отбора записей, к которым будет применена данная команда.
- Предложение WHERE не является обязательным.
- Упрощённый синтаксис:
 - ... **WHERE** condition ...
- Простые условия могут определяться:
 - операциями сравнения: `>`, `>=`, `<`, `<=`, `=`, `!=`, `<>`;
 - операциями выбора: `IN (list)`, `BETWEEN a AND b`, `LIKE (template)`;
 - логическими операциями: `NOT`, `IS NULL`.
- Составные условия определяются логическими операциями:
 - `AND`, `OR`.



Выражение WHERE

□ Примеры:

1. ... **WHERE** `id > 12;`
2. ... **WHERE** `surname = 'Ivanov';`
3. ... **WHERE** `comments IS NULL;`
4. ... **WHERE** `id BETWEEN 12 AND 112;`
5. ... **WHERE** `surname LIKE '_as%';`
6. ... **WHERE** `country IN ('RU' , 'UK' , 'US')`
`AND price > 1000;`



Команда UPDATE

- Упрощённый синтаксис:
 - **UPDATE** table **SET** column = value, ... **WHERE** condition;
- Пример:
 1. **UPDATE** a **SET** comment = '-' ;
 2. **UPDATE** a **SET** title = 'new record' **WHERE** id = 2 ;



Команда DELETE FROM

- Упрощённый синтаксис:
 - **DELETE FROM** table-name **WHERE** condition;
- Пример:
 1. **DELETE FROM a;**
 2. **DELETE FROM a WHERE id = 3;**



Команда SELECT

□ Упрощённый синтаксис:

□ **SELECT** ALL|DISTINCT *field-list* | * | *function*
FROM *table*

WHERE *condition*

HAVING *condition*

GROUP BY *field-list*

ORDER BY *field-list* ASC | DESC;



Выражения GROUP BY и HAVING

- Выражение
 - ... **GROUP BY** `field-list` ...
 - позволяет сгруппировать в одну запись записи с равными значениями полей, имена которых заданы в списке `field-list`.
- Действие выражения
 - ... **HAVING** `condition` ...
 - аналогично действию выражения `WHERE`, но применяется только к сгруппированным записям.



Выражение ORDER BY

- Выражение

... **ORDER BY** `field-list` ASC | DESC

позволяет упорядочить (отсортировать) записи в выборке по значениям полей, указанных в списке `field-list`.

- Сортировка записей в выборке выполняется в порядке перечисления полей в списке. По умолчанию применяется сортировка по возрастанию значений – ASC, DESC означает сортировку по убыванию значений.
- Порядок сортировки может быть указан для каждого поля в отдельности.



Примеры команды SELECT

- ❑ `SELECT * FROM a;`
- ❑ `SELECT * FROM a WHERE id = 1;`
- ❑ `SELECT * FROM a WHERE title LIKE '_s%';`
- ❑ `SELECT COUNT(*) AS Num, title FROM a GROUP BY title;`
- ❑ `SELECT COUNT(*) AS Num, title FROM a
GROUP BY title HAVING COUNT(*) > 1
ORDER BY Num DESC;`



Выражение JOIN

- Выражение
 - **INNER | LEFT | RIGHT | CROSS | NATURAL JOIN**
- позволяет объединить в одну запись записи из разных таблиц.
- Упрощённый синтаксис:
 - **SELECT ... FROM left-table**
INNER | LEFT | RIGHT | CROSS JOIN right-table
ON (condition)...
- где **left-table** – имя первой (левой) таблицы,
- **right-table** – имя второй (правой) таблицы,
- **condition** – условие объединения записей из таблиц.



Выражение JOIN

□ Примеры:

```
1. SELECT a.ID, a.TITLE, b.COMMENT FROM a  
   INNER JOIN b ON (a.ID = b.ID_A);
```

```
2. SELECT a.ID, a.TITLE, b.COMMENT FROM a  
   RIGHT JOIN b ON (a.ID = b.ID_A);
```

```
3. SELECT a.ID, a.TITLE, b.COMMENT FROM a  
   LEFT JOIN b ON (a.ID = b.ID_A);
```

```
4. SELECT a.ID, a.TITLE, b.COMMENT FROM a  
   NATURAL JOIN b;
```



Подмножество команд DCL

- **Data Control Language (DCL)** — подмножество языка SQL, команды которого предназначены для управления конфигурацией базы данных и управления пользователями.
- Некоторые команды подмножества:
 - **GRANT** — команда определения прав пользователя;
 - **REVOKE** — команда отмены прав пользователя.



Заключение

- Обзор рассмотренных тем
- Вопросы