

Онлайн образование

otus.ru



Проверить, идет ли запись

Меня хорошо видно && слышно?



Тема вебинара

DML: вставка, обновление, удаление, выборка данных



Коробков Виктор

Консультант команды технологического обеспечения
ООО «ИТ ИКС5 Технологии»

Telegram: @Korobkov_Viktor



Преподаватель



Виктор Коробков

более 20 лет в IT

специализация: проектирование баз данных (СУБД PostgreSQL, MS SQLServer)

В OTUS веду занятия на курсах: СУБД, PostgreSQL, SQL Server Developer, noSQL, Программист C

Правила вебинара



Активно
участвуем



Off-topic обсуждаем
в Telegram



Задаем вопрос
в чат или голосом



Вопросы вижу в чате,
могу ответить не сразу

Условные обозначения



Индивидуально



Время, необходимое
на активность



Пишем в чат



Говорим голосом



Документ



Ответьте себе или
задайте вопрос

Маршрут вебинара



SELECT: JOIN и WHERE

INSERT

UPDATE

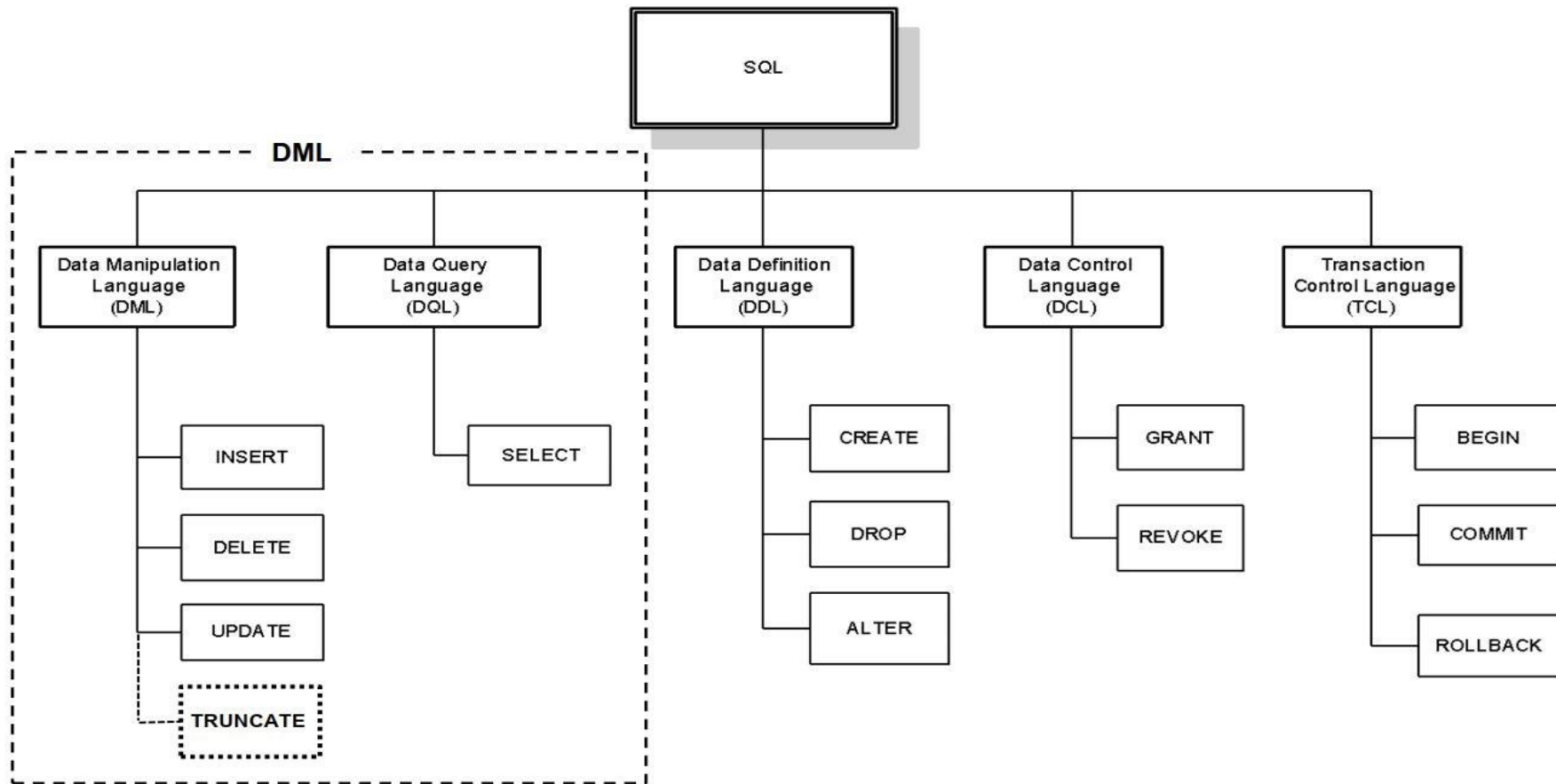
DELETE

Цели вебинара

После занятия вы сможете

1. Создавать различные типы связей между таблицами
2. Добавлять и обновлять данные со сложными выборками
3. Удалять данные с подзапросами

Структура SQL

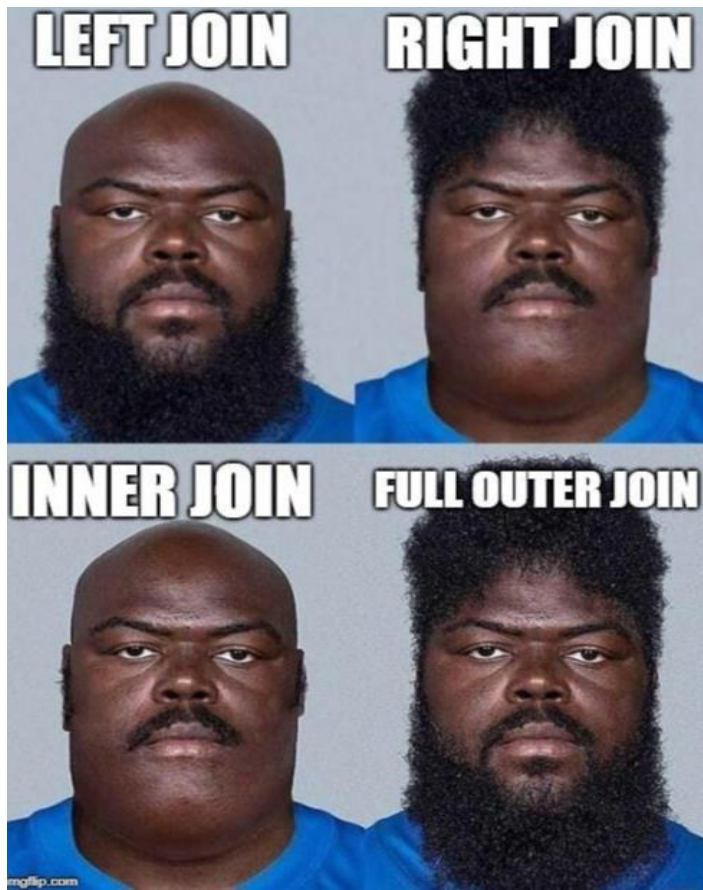


Соединение таблиц

table_1 *тип_соединения* **table_2** [*условие_соединения*]

Типы соединений

1. CROSS JOIN
2. INNER JOIN
3. OUTER JOIN
 - 3.1 LEFT JOIN
 - 3.2 RIGHT JOIN
 - 3.3 FULL JOIN



Декартово произведение

Все варианты сочетания строк из первой и второй таблицы:

```
FROM table_1 CROSS JOIN table_2
```

или

```
FROM table_1 INNER JOIN table_2 ON TRUE
```

или

```
FROM table_1, table_2
```

!!! осторожно при соединении трех и более таблиц

Соединение с сопоставлением строк

INNER JOIN – строки, которые есть в обеих таблицах



LEFT JOIN – все строки из левой и совпадающие строки из правой таблицы

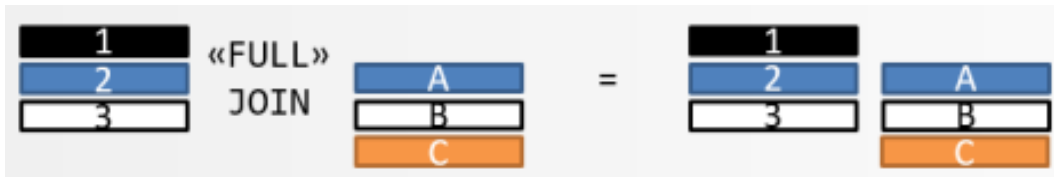


Соединение с сопоставлением строк

RIGHT JOIN – все строки из правой и совпадающие строки из левой



FULL JOIN – все строки из обеих таблиц, строки с совпадающими значениями формируют единый набор



Соединение с сопоставлением строк

INNER - внутреннее соединение

OUTER - внешнее соединение

- указание слов INNER и OUTER не обязательно;
- по умолчанию INNER соединение;
- при указании LEFT, RIGHT и FULL - OUTER соединение.

Условие соединения

T1 { [INNER] | { LEFT | RIGHT | FULL } [OUTER] }

JOIN T2 **ON** логическое_выражение

T1 { [INNER] | { LEFT | RIGHT | FULL } [OUTER] }

JOIN T2 **USING** (список столбцов соединения)

T1 **NATURAL** { [INNER] | { LEFT | RIGHT | FULL } [OUTER] }

JOIN T2

Условие соединения

ON - задает выражение логического типа, по аналогии с WHERE. Строки из таблиц соответствуют друг другу, если выражение ON возвращает для них true.

USING - сокращенная запись условия, применяется когда с обеих сторон соединения столбцы имеют одинаковые имена (исключается избыточность столбцов).

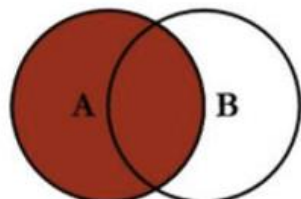
NATURAL - сокращённая форма USING: образует список USING из всех имен столбцов, существующих в обеих входных таблицах. Если столбцов с одинаковыми именами нет, то выполняется декартово произведение строк.

WHERE условие

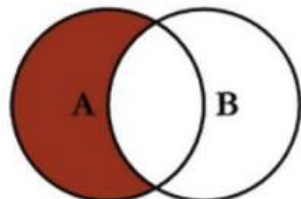
- >, <, =, != или <>
- BETWEEN начало AND конец диапазона (с включением!)
- IS NULL или IS NOT NULL
- [поле] [оператор] ANY|SOME|ALL (подзапрос)
- AND OR
- IN
- EXISTS
- LIKE
- SIMILAR TO
- Регулярные выражения

Типы соединения

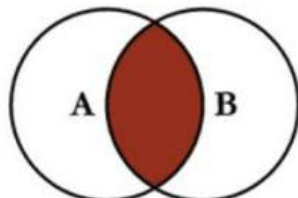
SQL JOINS



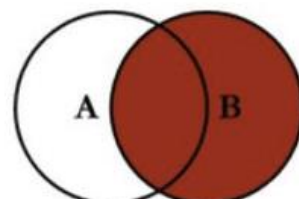
```
SELECT <select_list>  
FROM TableA A  
LEFT JOIN TableB B  
ON A.Key = B.Key
```



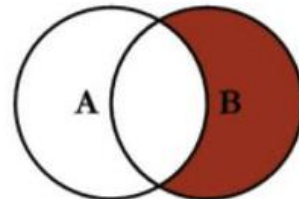
```
SELECT <select_list>  
FROM TableA A  
LEFT JOIN TableB B  
ON A.Key = B.Key  
WHERE B.Key IS NULL
```



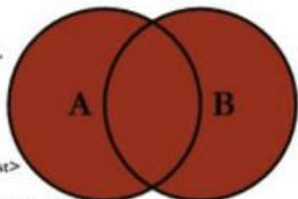
```
SELECT <select_list>  
FROM TableA A  
INNER JOIN TableB B  
ON A.Key = B.Key
```



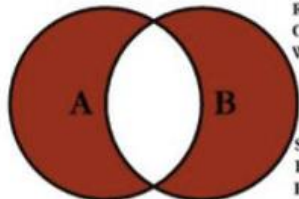
```
SELECT <select_list>  
FROM TableA A  
RIGHT JOIN TableB B  
ON A.Key = B.Key
```



```
SELECT <select_list>  
FROM TableA A  
RIGHT JOIN TableB B  
ON A.Key = B.Key  
WHERE A.Key IS NULL
```



```
SELECT <select_list>  
FROM TableA A  
FULL OUTER JOIN TableB B  
ON A.Key = B.Key
```



```
SELECT <select_list>  
FROM TableA A  
FULL OUTER JOIN TableB B  
ON A.Key = B.Key  
WHERE A.Key IS NULL  
OR B.Key IS NULL
```

Рекомендация

Чем проще запрос, тем ЛУЧШЕ)

INSERT INTO

```
INSERT INTO table (column1, column2, column3)
VALUES (value1, value2, value3);
```

```
INSERT INTO table (column1, column2, column3)
SELECT column1, column2, column3
FROM table2
```

Select INTO or Create Table AS

```
SELECT column_list  
      INTO [ TEMPORARY | TEMP ] [ TABLE ] new_table_name  
FROM table_name  
      WHERE condition;
```

```
CREATE TABLE table_name (column_name [, ...] )  
      AS query
```

UPDATE

UPDATE table

SET column1 = value1

WHERE column2 = value2;

UPDATE table1

SET column1 = table2.value1

FROM table1 JOIN table2 ON table1.id = table2.id

WHERE table1.column2 = value2

DELETE

```
DELETE FROM table1
```

```
    WHERE column1 = value1;
```

```
DELETE FROM table1 A
```

```
    USING table2 B
```

```
    WHERE B.id = A.id AND B.price > 10;
```

```
DELETE FROM table1
```

```
    RETURNING *;
```

Рефлексия



1. Отметьте, что запомнились с вебинара
2. Что будете применять в работе из сегодняшнего вебинара

**Заполните, пожалуйста,
опрос о занятии
по ссылке в чате**

Спасибо за внимание!

Приходите на следующие вебинары



Коробков Виктор