



ОНЛАЙН-ОБРАЗОВАНИЕ


Онлайн-образование

Не забыть включить запись!





Меня хорошо видно && слышно?

Ставьте  , если все хорошо
Напишите в чат, если есть проблемы

Тема “DML: вставка, обновление, удаление, выборка данных” Курс “СУБД”



Коробков Виктор

Правила вебинара



Активно участвуем



Задаем вопрос в чат или голосом



Off-topic обсуждаем в Slack



Вопросы вижу в чате, могу ответить не сразу

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

Смысл | Зачем это уметь

1 Для формирования оптимальных запросов по нескольким таблицам

2 Для редактирования данных средствами SQL

Маршрут вебинара

SELECT: JOIN и WHERE



INSERT



UPDATE



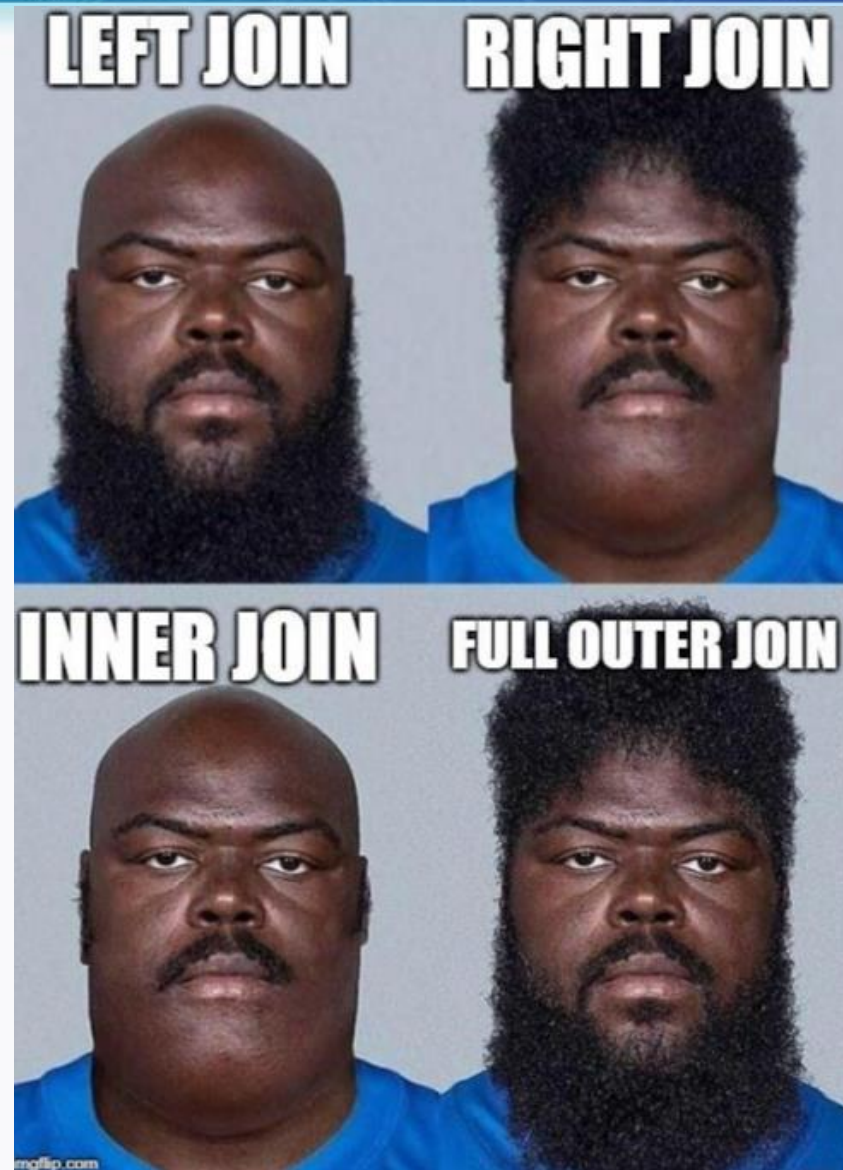
DELETE

Соединение таблиц

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

Типы соединения

1. CROSS JOIN
2. INNER JOIN
3. OUTER JOIN
4. LEFT JOIN
5. RIGHT JOIN
6. 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
```

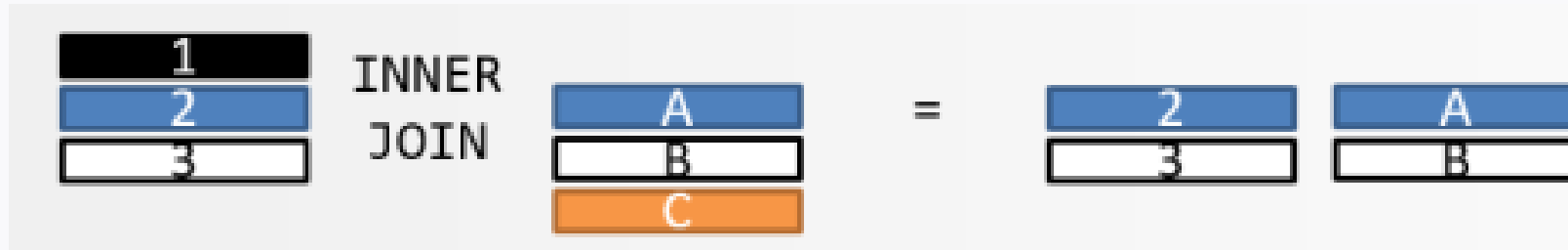
или

```
FROM table_1, table_2
```

!!! осторожно при соединении трех и более таблиц

Соединение с сопоставлением строк

INNER JOIN – строки, которые есть в обеих таблицах

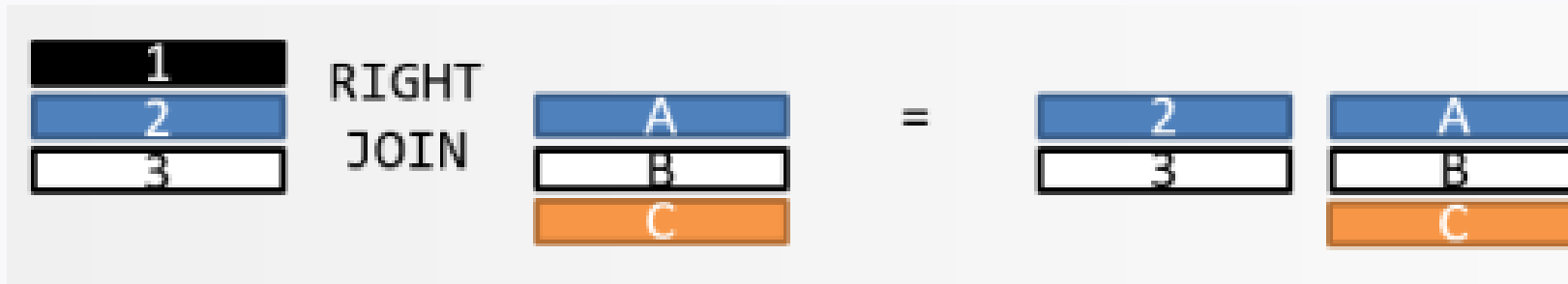


LEFT JOIN – все строки из левой и совпадающие строки из правой таблицы

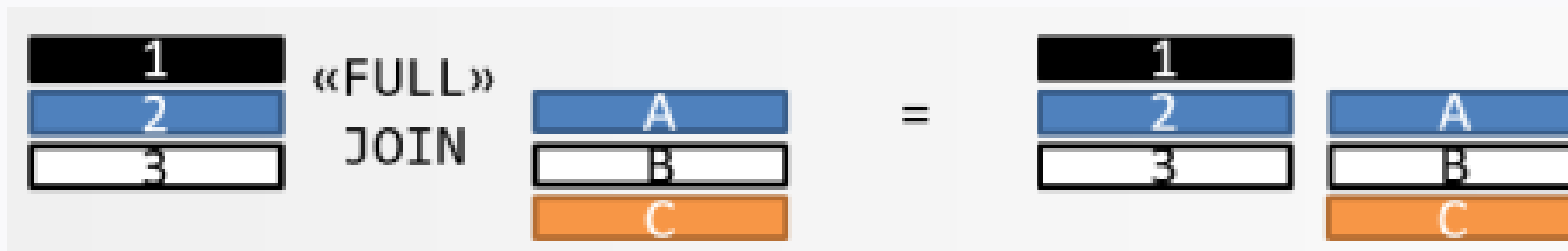


Соединение с сопоставлением строк

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).

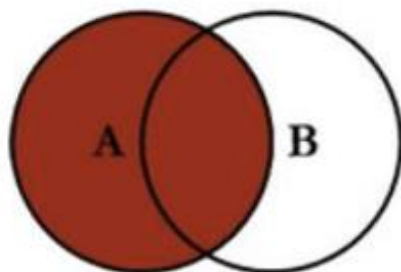
NATURAL - сокращённая форма USING: образует список USING из всех имен столбцов, существующих в обеих входных таблицах. Если столбцов с одинаковыми именами нет, то выполняется декартово произведение строк.

WHERE условие

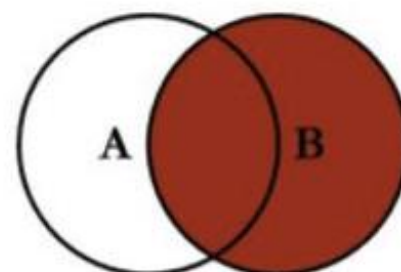
- >, <, =, != или <>
- BETWEEN начало AND конец диапазона (с включением!)
- IS NULL или IS NOT NULL
- [поле] [оператор] ANY|SOME|ALL (подзапрос)
- AND OR
- IN
- EXISTS
- LIKE
- SIMILAR TO
- Регулярные выражения

Типы соединения

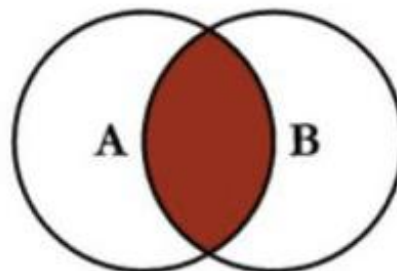
SQL JOINS



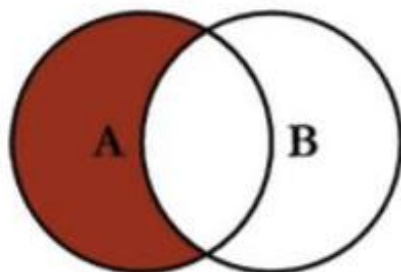
```
SELECT <select_list>  
FROM TableA A  
LEFT JOIN TableB B  
ON A.Key = B.Key
```



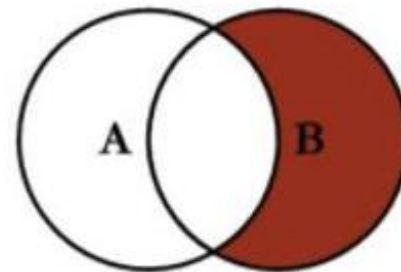
```
SELECT <select_list>  
FROM TableA A  
RIGHT JOIN TableB B  
ON A.Key = B.Key
```



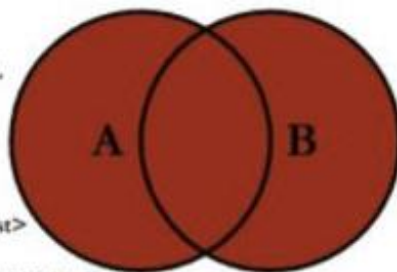
```
SELECT <select_list>  
FROM TableA A  
INNER JOIN TableB B  
ON A.Key = B.Key
```



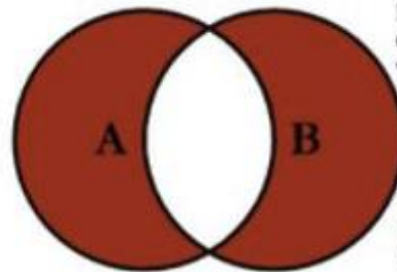
```
SELECT <select_list>  
FROM TableA A  
LEFT JOIN TableB B  
ON A.Key = B.Key  
WHERE B.Key IS NULL.
```



```
SELECT <select_list>  
FROM TableA A  
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.
```

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 alias1

SET column1 = alias2.value1

FROM table1 as alias1

JOIN table2 as alias2 ON table1.id = table2.id

WHERE alias1.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 *;
```

Домашнее задание

1. Напишите запрос по своей базе с регулярным выражением, добавьте пояснение, что вы хотите найти.
2. Напишите запрос по своей базе с использованием LEFT JOIN и INNER JOIN, как порядок соединений в FROM влияет на результат? Почему?
3. Напишите запрос на добавление данных с выводом информации о добавленных строках.
4. Напишите запрос с обновлением данные используя UPDATE FROM.
5. Напишите запрос для удаления данных с оператором DELETE используя join с другой таблицей с помощью using.

Рефлексия



Отметьте 3 пункта, которые вам запомнились с вебинара



Что вы будете применять в работе из сегодняшнего вебинара?

The background of the entire image is an aerial photograph of a city with many skyscrapers, overlaid with a semi-transparent blue layer. A network of white lines connects various points across the blue area, creating a digital or technological aesthetic.

Заполните, пожалуйста,
опрос о занятии по ссылке в чате

Спасибо за внимание!
Приходите на следующие вебинары



Коробков Виктор