

Занятие 3

QA.

START UP

Сложные выборки данных из таблиц

QA.

START UP

Темы занятия

1 Понятие ключа. Первичный и внешний ключи

2 Подзапросы

3 Декартово произведение

4 Соединения (оператор JOIN) и их типы

5 Сортировка набора данных и оператор TOP

6 Объединение выборок (оператор UNION)

7 Агрегация данных в запросах

Типы связей и ограничения

- Первичный ключ. Понятие уникальности кортежа. Простой и составной ключ
- Суррогатный ключ
- Внешние ключи. Ссылочная целостность
- Проверочные ограничения
- Понятие неопределенного значения (NULL)

Сложные выборки и варианты их осуществления

- Подзапрос (subquery)
 - в столбцах (секция **SELECT**)
 - в секции **FROM**
 - в секции **WHERE**
- Декартово произведение + предикат по равенству значений полей
- Оператор **JOIN**
 - Перекрестное объединение **CROSS JOIN**
 - Внутреннее объединение **INNER JOIN**
 - Внешнее объединение **OUTER JOIN**:

LEFT OUTER JOIN | RIGHT OUTER JOIN

- Полное объединение **FULL JOIN**

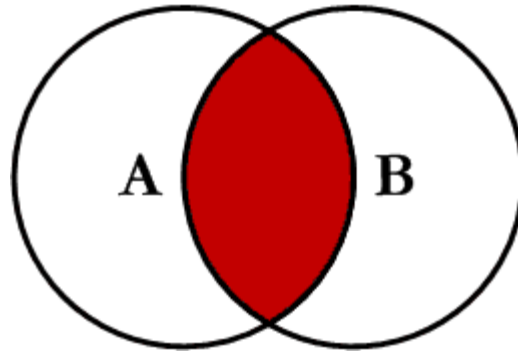
Декартово произведение, CROSS JOIN

```
SELECT <select_list>  
FROM    TableA, TableB
```

```
SELECT <select_list>  
FROM    TableA  
          CROSS JOIN TableB
```

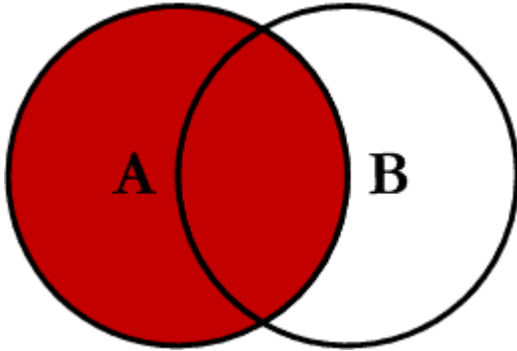
```
SELECT <select_list>  
FROM    TableA A, TableB B  
WHERE    A.KeyID = B.KeyID
```

INNER JOIN



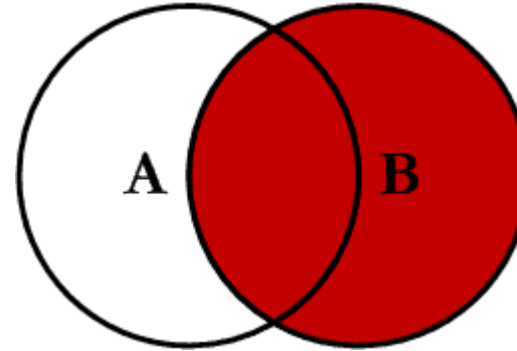
```
SELECT <select_list>  
FROM   TableA A  
        INNER JOIN TableB B  
        ON   A.KeyID = B.KeyID
```

LEFT и RIGHT OUTER JOIN



LEFT OUTER JOIN

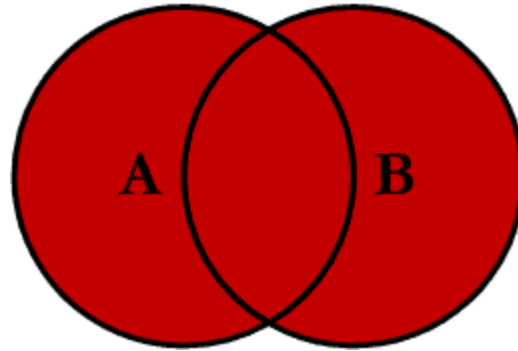
```
SELECT <select_list>  
FROM   TableA A  
        LEFT OUTER JOIN TableB B  
        ON A.KeyID = B.KeyID
```



RIGHT OUTER JOIN

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

FULL OUTER JOIN



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


Сортировка набора данных и оператор TOP

Предложение **ORDER BY** сортирует данные, возвращаемые запросом

- Упорядочение результирующего набора запроса
- Определение порядка сортировки

Синтаксис **ORDER BY**::=

```
ORDER BY order_by_expression [COLLATE  
collation_name] [ASC | DESC] [, ...n]
```

Оператор **TOP** ограничивает число строк, возвращаемых в результирующем наборе запроса до заданного числа или процентного значения. Используется совместно с предложением **ORDER BY**

Синтаксис **TOP**::=

```
[TOP (expression) [PERCENT] [WITH TIES]]
```

Объединение выборок. UNION и UNION ALL

- Объединяет результаты двух или более запросов в один результирующий набор
- Количество и порядок столбцов должны быть одинаковыми во всех запросах
- Типы данных должны быть совместимыми
- **ALL** объединяет в результирующий набор все строки (вместе с дубликатами)

Синтаксис **UNION [ALL]::=**

```
{<query_specification> | (<query_expression>)}  
  UNION [ALL]  
  <query_specification | (<query_expression>)  
  [UNION [ALL] <query_specification> |  
  (<query_expression>)  
  [ ...n ]]
```

Агрегация данных в запросах

- Оператор **GROUP BY**
- Фильтрация агрегированных данных с помощью **HAVING**
- Основные функции агрегации
 - **COUNT**
 - **MAX, MIN**
 - **SUM**
 - **AVG**

Синтаксис **GROUP BY::=**

```
SELECT    <select_list>, <aggregate_func>
FROM      <table_name>
WHERE      <search_condition>
GROUP BY  <select_list>
HAVING    <search_condition>
```

СПАСИБО

