

Вспомогательное Select

[SELECT column 1, column 2, ...
FROM table_name;

* = все rows

- result-set = result table that is returned

Вспомогательное WHERE

- возвращает только строки, соответствующие условию

[SELECT column 1, ...
FROM table
WHERE condition

DQL: SQL Data Query Language

SELECT
select_list

FROM
source

Optional [WHERE
search-condition

GROUP BY
grouping expression

HAVING
search condition for groups

ORDER BY
order expression]

SELECT

- * = все поле возвратить. Не рекомендуется использовать. Потому что (1) можно добавиться ненужные поля, (2) сложнее найти нужное поле, (3) может вернуться слишком объемный ответ.

- SELECT table_name.*
- SELECT [table_name.] column_name
[AS alias_name]
- SELECT expression [AS alias_name]
- может быть несколько условий, разделенных запятой
вычисление на стороне сервера

FROM:

- table_name | view_name | function
[AS alias]
рекомендуется всегда использовать alias - чтобы не путать поля: таблица. поле

• источников и быть несколько -
разделяются запятой

• Все таблицы и представления
размещаются в схемах

• CASE - императивные выражения

SELECT ...
CASE

WHEN expression THEN expression

[...n]

[ELSE expression]

END

FROM ...

• BETWEEN

WHERE column-name BETWEEN value AND value

• IN WHERE column-name IN (values)

• NOT IN

WHERE column-name NOT IN (values)

• LIKE - выборка по шаблону строки

WHERE col-name LIKE 'template'

% - любые символы
_ - 1 символ

NULL VALUE

- ~~пустая ячейка = NULL~~
- NULL = unknown value
- NULL - это не 0
- NULL - это не пустая строка
- Операции с NULL - это 3-valued logic
- Проверка ячейки: IS[NOT]NULL
- Сравнение с NULL → NULL

ORDER BY

SELECT...

FROM...

ORDER BY col_name [DESC|ASC] [...n]

TOP

SELECT TOP (number)

FROM...

- выбор первых n строк

DISTINCT

SELECT DISTINCT col_name [...n]
FROM table_name

- выбор уникальных значений.

При выборе нескольких

DISTINCT- столбцов,

уникальность считается
по комбинации значений.

JOINS - multitable queries

SYNTAX:

FROM

{

table_source_1

[inner/left/right/full outer]

но упоминаю

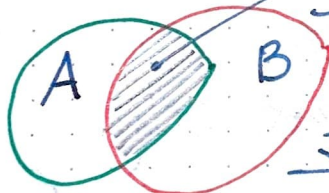
JOIN

table_source_2

ON condition

}

Inner Join



значения, где
которые удовлетво-
ряют условию

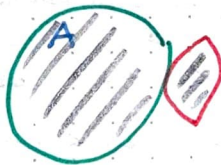
JOIN



LEFT JOIN:



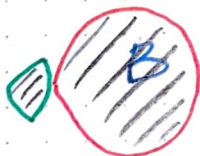
JOIN



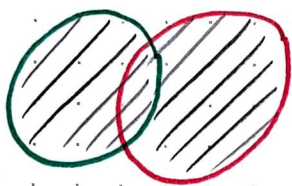
RIGHT JOIN:



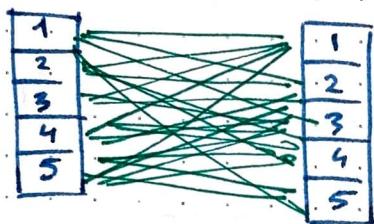
JOIN



FULL JOIN:



CROSS JOIN - декартово произведение:



Grouping and aggregation

Column functions

- SELECT SUM(col-name)
FROM table-name

- SUM, AVG, MIN, MAX, COUNT

Group By.

not NULL!

- В результате - выборка - каждая строка соответствует отдельному значению колонки из 'group by'

- В SELECT могут быть: колонки из GROUP BY и агрегирующие функции

• HAVING clause

- Включает в выборку только соответствующие условиям значения.

- В HAVING можно использовать только столбцы / выражения из SELECT.

• SUBQUERIES - подзапросы

- Например, select внутри select.
 - Внутренний select - источник данных для внешнего select.

- Например, select внутри where.
 - Внутри выраж. сравнения ^{пример:}
`where col_name >= (select max(col_name) from table_name)`

- Внутри 'IN' - membership test
- Внутри 'EXISTS' - existence test
 - TRUE - если возвращается хотя бы одна строка
- Внутри ANY / ALL
ALL / ANY: сравнение со всеми записями в результате подзапроса. ANY - какое-либо знач.; ALL - все значения