

# Oracle Core

## Тема 3

### **DML**

(часть 1)

# Содержание

- ☐ История SQL
- ☐ Оператор SELECT
- ☐ Соединения таблиц (joins)
- ☐ Подзапросы
- ☐ Set операторы
- ☐ Группировка и агрегатные функции
- ☐ Условия ANY, ALL, SOME
- ☐ Условие IN
- ☐ Условия LIKE и REGEXP\_LIKE
- ☐ Условие EXISTS
- ☐ Условие BETWEEN
- ☐ Выражение CASE

# История SQL

- ❑ СУБД System R (компания IBM, начало 1970-х) – язык SEQUEL (Structured English Query Language)
- ❑ 1983 год, начало разработки стандарта организациями ISO и ANSI
- ❑ 1986 год, SQL-86. Первая версия стандарта
- ❑ 1989 год, SQL-89
- ❑ 1992 год, SQL-92
- ❑ 1999 год, SQL:1999
- ❑ 2003 год, SQL:2003
- ❑ 2006 год, SQL:2006
- ❑ 2008 год, SQL:2008
- ❑ 2011 год, SQL:2011

# Группы команд SQL

- ❑ DDL – Data Definition Language – команды ddl позволяют изменять структуру базы данных (create, alter, drop)
- ❑ DML – Data Manipulation Language – команды dml позволяют извлекать или изменять содержимое базы данных (то есть сами данные)
- ❑ DCL – Data Control Language – команды dcl позволяют разграничивать права пользователя (grant, deny, revoke)
- ❑ TCL – Transaction Control Language – команды tcl позволяют управлять транзакцией (set transaction, commit, rollback)
- ❑ К DML командам относят, в частности, следующие команды: **Select, Insert, Update, Delete, Merge, Explain Plan, Lock Table**

# Оператор select

Select – это единственный sql-оператор, который используется при запросе данных.

Минимально возможная структура запроса выглядит следующим образом:

```
Select <some column>
```

```
From <some table>
```

Ключевые слова **select** и **from** присутствуют в любом запросе

# Структура оператора select

**Select** [All | Distinct ] <[Table. | TableAlias.]<\* | Column> [ColumnAlias], ....

**From** Table1 [Table1Alias], Table2 [Table2Alias]

[**Where** <conditions>]

[**Order by** <Column1 | Column1Alias>, ... [Asc | Desc]]

# Prerequisites to select execution

1. Для выполнения запроса `select` к таблице или представлению, пользователю необходимо иметь привилегию `select` на эту таблицу или представление
2. Пользователю-владельцу схемы, в которой лежит представление, необходимо иметь привилегии `select` на все таблицы из этого представления

# Joins

- ✓ Inner join
- ✓ Outer join
  - Left outer join
  - Right outer join
  - Full outer join
- ✓ Cartesian product
- ✓ Self join
- ✓ Partitioned outer join
- ✓ Antijoin
- ✓ Semijoin

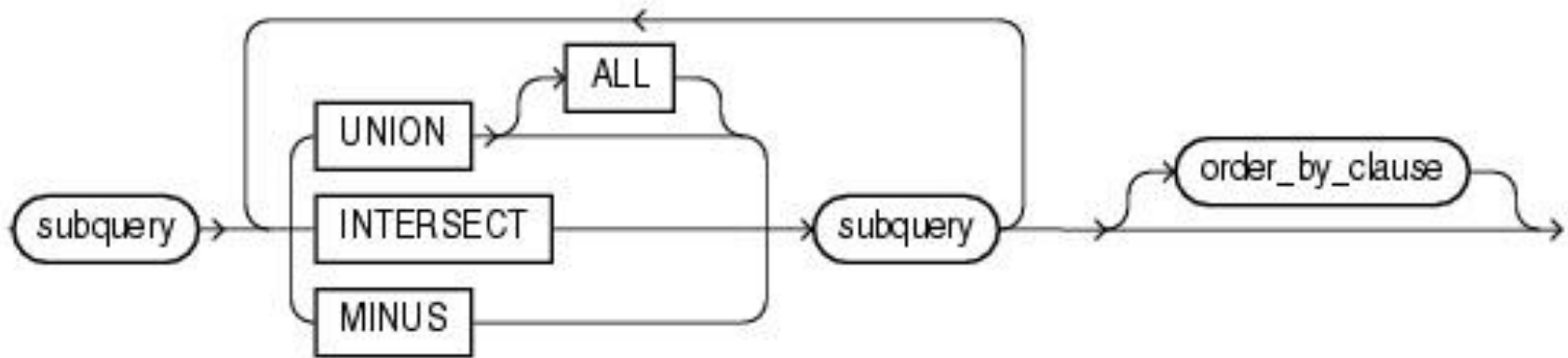


# Подзапросы (subqueries)

Подзапрос (subquery) – это select-запрос, вложенный в другой sql-запрос.

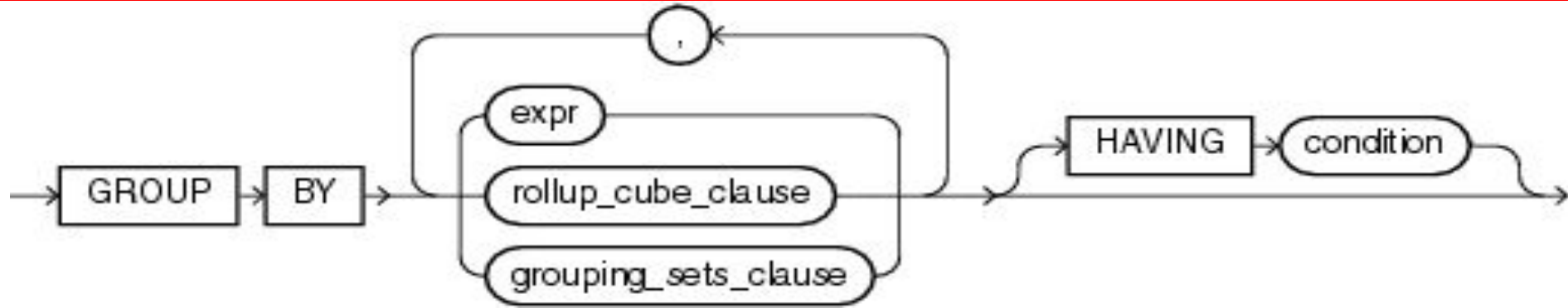
- ✓ Подзапрос в секции select
  - Видит таблицы из внешнего from
  - Может выбирать не более одной записи и одного значения
- ✓ Подзапрос в секции from
- ✓ Подзапрос в секции where
  - Видит таблицы из внешнего from
  - Для операции “=” должен возвращать только одну строку и одно значение
  - Для операции in должен возвращать только одно значение (одно поле), но произвольное количество записей (или не возвращать ничего)

# Set operators



- ✓ Union All – возвращает объединение запросов
- ✓ Union – после объединения запросов делает distinct
- ✓ Intersect – возвращает пересечение запросов
- ✓ Minus – вычитает результаты одного запроса из результатов другого

# Группировка



✓ Group by **clause**

✓ Group by **cube**

✓ Group by **rollup**

✓ Group by **grouping sets**

# Агрегатные функции

- ✓ Возвращают агрегированное значение по набору данных
- ✓ Могут использоваться в секциях select, having и order by
- ✓ Sum
- ✓ Min
- ✓ Max
- ✓ Count
- ✓ Avg
- ✓ All | Distinct
- ✓ [Создание собственных агрегатных функций](#)

# Условия any, all и some

expr

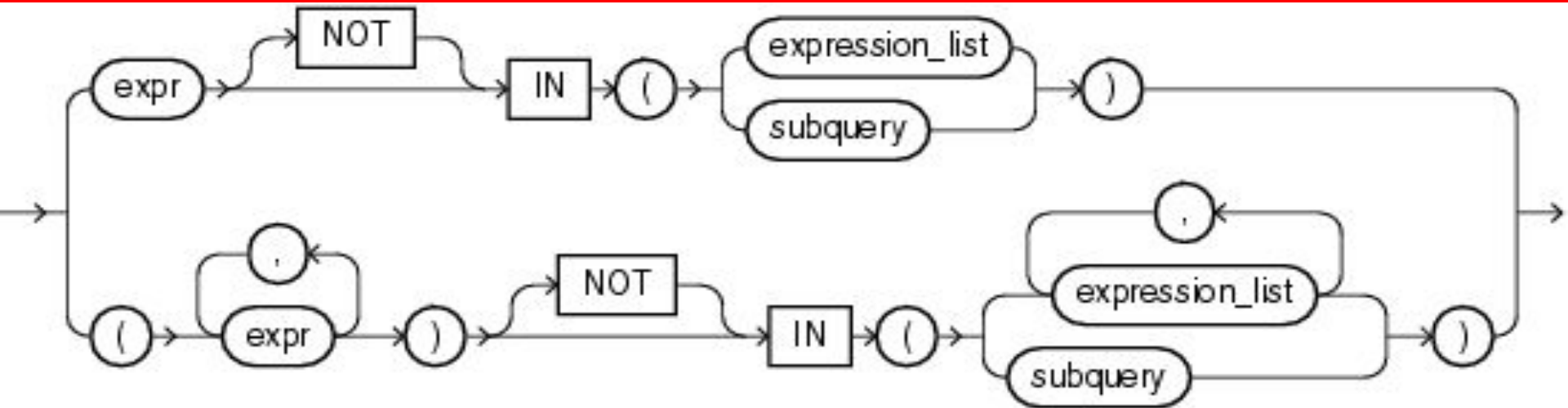
{ = | != | ^= | <> | > | < | >= | <= }

{ ANY | SOME | ALL }

( { expression\_list | subquery } )

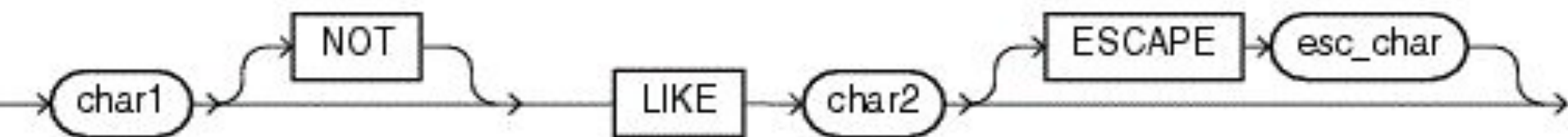
- ✓ ANY SOME – возвращает true, если операция истинна для одного или больше значений из expression\_list, иначе false
- ✓ ALL – возвращает true, если операция истинна для всех значений из expression\_list, иначе false

# Условие IN



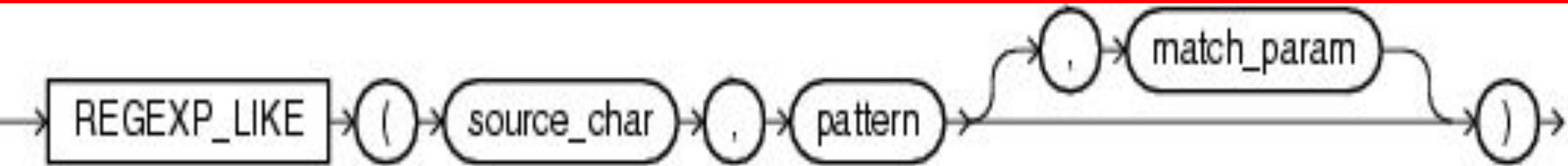
- ✓ IN – эквивалентен =ANY
- ✓ NOT IN – эквивалентен !=ALL
- ✓ Antijoin
- ✓ Можно указать до 1000 значений/выражений в expression\_list
- ✓ Эти выражения не всегда выполняются в том порядке, в каком они указаны в expression\_list
- ✓ Если в expression\_list(или результате subquery) оператора NOT IN есть значение null, результат оператора всегда будет false

# Условие LIKE



- ✓ Позволяет проверить строку на соответствие шаблону
- ✓ Знак подчеркивания «\_» соответствует любому одному символу
- ✓ Знак процента «%» соответствует любой строке, в том числе пустой
- ✓ Выражения char1, char2 и esc\_char могут иметь один из следующих типов данных: CHAR, VARCHAR2, NCHAR ИЛИ NVARCHAR2. Если типы данных у выражений отличаются, Oracle неявно приводит все выражения к типу данных char1

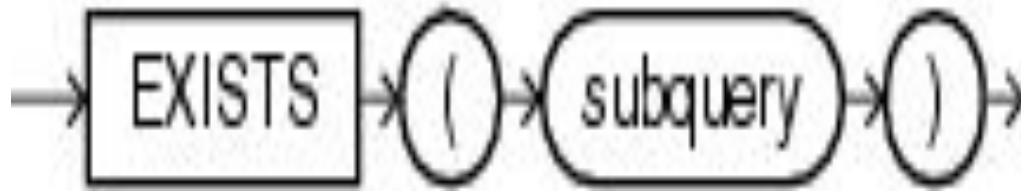
# Условие REGEXP\_LIKE



- ✓ source\_char – строка (как правило, строковое поле таблицы), которую сравниваем с шаблоном
- ✓ pattern – регулярное выражение (шаблон), максимальный размер 512 байт
- ✓ match\_param – параметры сравнения (i – регистронезависимое сравнение, s – регистрозависимое и некоторые другие)
- ✓ Синтаксис регулярного выражения (pattern) [ТУТ](#)

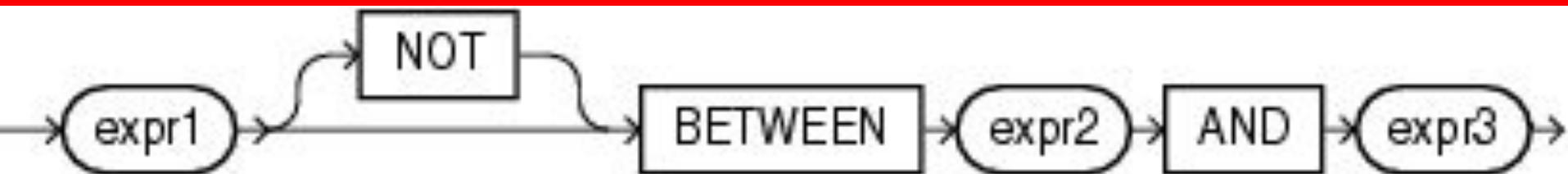


# Условие EXISTS



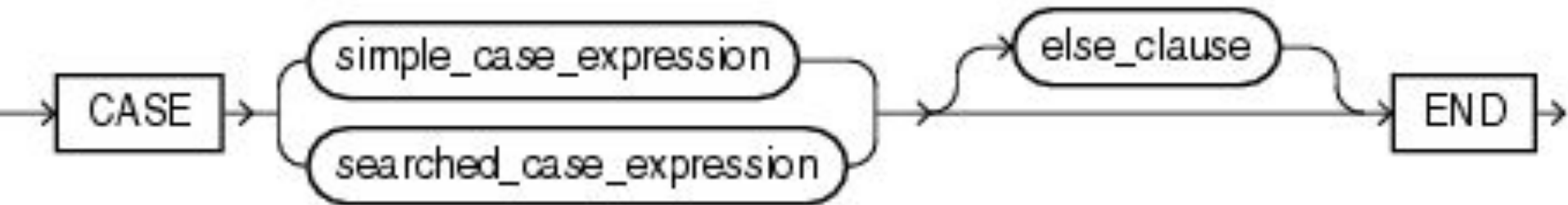
- ✓ Возвращает true, если результат подзапроса содержит хотя бы одну запись. Иначе false
- ✓ Semijoins

# Условие between

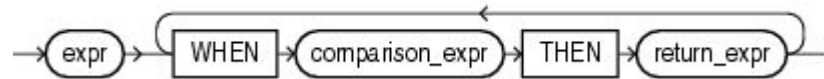


- ✓ Возвращает true, если значение выражения `expr1` входит в интервал `expr2` и `expr3` (включая границы). Т.е. эквивалентно такому условию: `expr2 <= expr1 AND expr1 <= expr3`
- ✓ Все три выражения должны иметь числовой, строковый или datetime тип
- ✓ Если результаты выражений `expr1`, `expr2` и `expr3` имеют разные типы данных, Oracle неявно производит преобразование к единому типу данных. В случае, если преобразование невозможно, генерируется исключение
- ✓ `expr1 between expr2 and expr3`, где `expr3 < expr2`, возвращает null при `expr1` равным null, и возвращает false при `expr1` не равным null
- ✓ `expr1 not between expr2 and expr3`, где `expr3 < expr2` и `expr1` не null, возвращает true

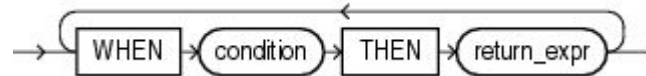
# Выражение case



✓ simple\_case\_expression



✓ searched\_case\_expression



✓ else\_clause → ELSE[ELSE] → else\_expr([else\_expr]) →

✓ Позволяет использовать if... then...else логику в sql-выражениях

✓ Для случая с searched\_case\_expression Oracle проверяет все условия слева направо до первого true, и возвращает соответствующий ему return\_expr. Если такого условия не находится, возвращается else\_expr. Если секция ELSE отсутствует, возвращается null.

✓ Допускает не более 65535 аргументов

✓ Для expr, comparison\_expr и return\_expr допускает следующие типы данных:

# Summarizing

- ☐ Оператор SELECT
- ☐ Соединения таблиц (joins)
- ☐ Подзапросы, set операторы
- ☐ Группировка и агрегатные функции
- ☐ Условия ANY, ALL, SOME
- ☐ Условие IN, LIKE и REGEXP\_LIKE, EXISTS, BETWEEN
- ☐ Выражение CASE

# Использованные материалы

1. [«Понимание SQL», Мартин Грубер](#)
2. [Select Statement](#) (oracle documentation)
3. [Joins](#) (oracle documentation)
4. [Group by](#) (oracle documentation)
5. [Aggregate Functions](#) (oracle documentation)
6. [Условие IN](#) (oracle documentation)
7. [Условие Exists](#) (oracle documentation)
8. [Условие Between](#) (oracle documentation)
9. [Выражение Case](#) (oracle documentation)
10. [Условие Like](#) (oracle documentation)

**Везде ссылки на документацию к версии Oracle 11g Release 2**