DML – те команды, которые работают с внутренней информацией (SELECT, INSERT, UPDATE,DELETE, MERGE)

DDL – с данными в таблицах не работают, а работают со структурой. Можно создавать таблицы с помощью CREATE, изменять таблицы с помощью Alter, удалять обьекты с помощью DROP, изменять имя столбцов с помощью RENAME, TRANCATE удаляет все данные

TCL – COMMIT, ROLLBACK, SAVEPOINT. Когда вставляются новые данные эти команды нужны для подтверждения окончательного изменения. Rollback откатывает если передумали.

DCL – grant, revoke. Дает права на просмотр или изменение данных

**Типы данных**

Number (5,1) где 5 – это всего максимальное количество цифр в числе, а 1 количество цифр в дробной части

Char(length) – текст c фикс. длиной length

Varchar2(size) – текст с переменной длиной максимальной длиной size

Date – инфо о годе, месяце, дне, часе , минуте, секунде

Timestamp – то же что и date только еще показывает доли секунды

Null – отсутствие данных ( 0 и пробел не null)

**Select statement (не меняет данные)**

**Концепция projection - Вывод определенных столбцов**

DESCRIBE Table\_name – показывает информацию о таблице

Select col1, col2 from table\_name

Select DISTINCT col1, col2 from table\_name – извлекает уникальные данные из таблицы. Уникальные комбинации столбцов col1 и col2.

**Expressions in select list**

SELECT col1, col2, expressions FROM table

SELECT salary\*2/(2+3) FROM employees; Для численных типов данных

SELECT end\_date-start\_date+1 from job\_history. Вычитает даты

SELECT start\_date+7 from job\_history. Когда прибавляем число к дате, получаем новую дату

SELECT first\_name || last\_name from table. Соединяет текст из двух столбцов в одном с помощью конкатенации.

SELECT ‘My first name is ‘ || first\_name from table

ALIAS – альтернативное имея для столбца или целого выражения. Новое название пишется сразу после название столбца или выражения

SELECT columns alias, expressions alies FROM table

SELECT 'My name is' || first\_name **name** FROM employees – после выражения или названия столбца пишем свое название

Select “Любой текст” или выражение from dual – dual таблица для вывода данных, которых нет в базе данных

select q'/It's my life/' song from dual – q ‘delimeter TEXT delimeter’

**Концепция selection - Вывод определенных строк**

Select columns from table Where conditions - выведи мне все столбцы из такой то таблицы, где будут удовлетворяться определенные условия

select \* from departments where location\_id IN (1700, 2400, 1500) – выводит всю информацию где ай ди локации удовлетворяет одной из списка в скобке

select \* from employees where commission\_pct IS NULL – выводит данные, для которых отсутствует выбранное значение

select first\_name from employees where first\_name like **‘s%’** – выбирает имена начинающиеся на S. Оператор LIKE помогает выбрать данные подходящие под определенную маску. % - заменяет любое количество символов. \_ - заменяет один любой символ

select \* from employees where job\_id like 'ST\\_%' ESCAPE '\' - escape ‘\’ говорит, что 1 символ после \ не будет считаться специальным.

select \* from employees where **NOT** job\_id = 'SA\_MAN';

select \* from employees where **NOT** **LIKE** 'SA\_MAN';

Приоритетность операторов – сначала выполняется операции в скобках AND в первую очереднь, OR во вторую.

select \* from employees

**order by** salary **desc** – сортировка по какому либо критерию от большего к меньшему. Без desc от меньшего к большему

select first\_name fn, salary sl from employees

where salary > 10000

order by salary desc, fn;

select first\_name, salary from employees order by 2 – сортирует по второму столбцу из выбранных столбцов

**ФУНКЦИИ**

**Single-row функции - сколько строк в таблице, столько итоговых строк и выведется**

**Multiply-row функции – из многих строк выводит только один результат**

**ТЕКСТОВЫЕ ФУНКЦИИ**

1. LOWER(s) - принимает строку или текст и выводит текст с уменьшенными буквами. Игнорирует числа.

Select first\_name, LOWER (first\_name) from employees;

2. UPPER(s) - принимает строку или текст и выводит текст c заглавными буквами. Игнорирует числа.

3. INITCAP(s) – принимает текст. В каждом слове текста делает первую букву заглавной, а остальные прописными. Любой знак пунктуации разделяет слова.

4. CONCAT(s1, s2) – соединяет два текста. Числа и даты конвертируются в текст и обьединяются.

5. LENGTH(s) - принимает строку и вычисляет ее длину.

6. LPAD(s,n,p) и RPAD – добавляет определенные символы слева и справа от строки. S – строка, n – конечная длина текста, p – текст для добавления.

7.TRIM (trailing/leading/both ‘что удалить’ from ‘откуда’) – удаляет символы с начала или с конца текста. Trailing удаляет все необходимые символы с конца, leading – c начала, both – с обоих сторон.

Select TRIM(‘ ARTEM ‘) form dual – удалит все пробелы сзади и спереди

select TRIM( '!' from '!!!HELLO!! MY FRIEND!!!!!!!!') from dual

8.INSTR (s, search\_string, start\_position, Nth occurance) – ищет позицию определенного текста в заданном тексте начиная со start\_position. Nth occurance – ищет определенный номер появляния искомого значения. Выводит номер первого символа, где появилось совпадение.

9. SUBSTR(s, start position, number of charactres) - выводит какое то количество символов текста начиная с определенной позиции

10. REPLACE(s , searh\_item, replacement\_item) – заменяет в тексте определнный текст search\_item на другой текст. Если другого текста нет, то из текста просто удаляется search\_item

**Числовые функции**

1. ROUND(n, precision) – до какой точности мы хотим округлить число

select ROUND(3453, -2) from dual – округлит последние 2 цифры до 3500

2. TRUNC(n , num) – просто обрезает дробную часть до количества знаков num

select TRUNC(3453, -2) from dual – отсечет последние 2 цифры до 3400

3. MOD (dividend, divisor) – находит остаток от деления делимого на делитель.

**Date функции**

1. SYSDATE – возвращает время нашего сервера или компьютера.

2. MONTHS\_BETWEEN (end\_date, start\_date) – выводит количество месяцев между конечной и начальной датой.

3. ADD\_MONTHS(date, n) – добавляет к дате n месяцев

4. NEXT\_DAY(day, day\_of\_the\_week) – количество дней до конкретного дня недели или ближайшую дату выбранного дня

5. ROUND(date, presion\_format). Век – CC, год – YYYY, четверть – Q, месяц – MM, неделя – WW, день – DD, час – HH, минута – MI

6. TRUNC(date, precision\_format). Обрубает дату до точности округления.

**CONVERSION, GENERAL, CONDITIONAL функции**

**1. TO\_CHAR(num, format mask, nls\_parameters) – преобразовывает число в текст в указанном формате.**

select to\_char(123423428, '9999999999') from dual; 9- заменяет собой одну цифру

select to\_char(18, '0999999') from dual; - ставит перед числом нули.

select to\_char(18.35, '099999.999') from dual;

select to\_char(1234567, '999,999,9999') from dual;

select to\_char(18, '$99') from dual; добавляет доллар перед числом

select to\_char(-18, '999MI') from dual; добавляет минус после числа

select to\_char(-18, 'S0999') from dual; добавляет минус перед числом

**2. TO\_CHAR (date, format\_mask, nls\_param) - преобразовывает дату в текст в указанном формате.**

select to\_char(sysdate, 'YY') from dual; - ‘Y’ – возвращает последнюю цифру или цифры года.

RR – год в формате двух цифр;

'Year'- выводит буквенное написание года

MM – месяц в формате двух цифр

MON – 3 буквы из названия месяца

MONTH – буквенное написание месяца

select hire\_date, to\_char(hire\_date, 'MONTH', 'NLS\_DATE\_LANGUAGE=RUSSIAN') from employees; - выводит полное название месяцов на русском

select hire\_date,

to\_char(hire\_date, 'fmMONTH', 'NLS\_DATE\_LANGUAGE=RUSSIAN') || ' hello!'

from employees; fm – убирает лишние пробелы с имен, дней и месяцев -

D – порядковый день недели (на англе считается с воскресенья)

DD – порядковый день месяца;

DDD – день года

DY – 3 буквы из названия дня недели

DAY – полное название дня недели

W – неделя месяца

WW – неделя года

Q – квартал года

СС – век

AM, PM – индикатор показываюший АМ или РМ в данный момент

HH:MI – часы и минуты в 12 часовом формате

HH24:MI-SS – часы и минуты и секунды в 24 часовом формате

SSSSS – секунда после полуночи

select to\_char(sysdate, 'DAY!MON?yyyy') from dual; - WEDNESDAY!JAN?2024

select to\_char(sysdate, 'DDth " of " Month') from dual; - 31ST of January

select to\_char(sysdate, 'yyyysp ') from dual; - преобразует в буквенное произношение

**3. TO\_DATE(text, format\_mask, nls-param) – преобразует текст в дату используя определенный формат**

select TO\_DATE('18:40 2019!17-Sep', 'HH24:MI yyyy!dd-MM ') from dual; выводит 17-SEP-19

**4. TO\_NUMBER(text, format\_mask, nls) – переводит текст в число по опредленному формату**

select TO\_NUMBER('$4555.77','$9999999.999') from dual; - выводит 4555.77

**5. NVL(commission\_pct, 0) - если искомое значение равно null, то выводит вместо него 0**

**6. NVL2(value, ifnotnull, ifnull) -** **если искомое значение не равно null, то выводит вместо него второй параметр, если искомое значение равно null, то возвращает третий параметр.**

**7. NULLIF(value1, value2)- если два параметра равны между собой, то возвращается Null, если не равны, то возвращается первый параметр**

**8. COALESCE(value1, value2, …, valueN) – возвращает первое значение, которое не null**

**9. DECODE (expr, comp1, iftrue1, comp2, iftrue2,…iffalse) – сравниваем значения comp1, comp2 и тд с expr и выводим iftrue1, если эти значения совпадают и iffalse, если не совпадают.**

**10. SIMPLE CASE**

CASE expr

WHEN comp1 THEN iftrue1

WHEN comp2 THEN iftrue2…

ELSE iffalse

END

**11. SEARCHED CASE**

CASE

WHEN cond1 THEN iftrue1

WHEN cond2 THEN iftrue2…

ELSE iffalse

END

**GROUP- функции. ORDER BY, HAVING**

1. COUNT(\* or DISTINCT col) from table – считает все строки удовлетворяющие условию. Не считает null.
2. SUM(num1,num2,…numN) – суммирует все значения
3. AVG(num1,num2,…numN) – считает среднее арифметическое значений
4. MAX()- выводит макс. Значение. Работает для всех типов данных
5. MIN() - выводит мин. Значение. Работает для всех типов данных

**GROUP BY**

SELECT job\_id, ROUND(AVG(salary)) avg\_sal, min(salary) min\_sal, max(salary) max\_sal

FROM employees

WHERE length(first\_name) > 4 and salary > 5000

GROUP BY job\_id - *нельзя группировать по alias*

ORDER BY avg\_sal – *можно упорядочивать по alias, по номеру и по выражению*

select \* from employees;

select department\_id, job\_id, count(\*) from employees

GROUP BY department\_id, job\_id – *группирует по двум атрибутам*

ORDER BY 1;

select job\_id, to\_char(hire\_date, 'yyyy') year, sum(salary)

from employees

where job\_id in ('ST\_CLERK', 'SA\_REP') and employee\_id > 115

group by job\_id, to\_char(hire\_date, 'yyyy')

**HAVING**

select department\_id, count(\*) emp\_num from employees

WHERE LENGTH(first\_name) > 4

group by department\_id

HAVING count(\*) > 3 – having отбирает сгрупированные данные по определенному условию. Работает только на группах. Where работает с существующими строками.

order by department\_id;

**JOIN**

select e.salary, e.first\_name from employees e – алиас для таблицы

1. **NATURAL JOIN** – обьединяет таблицы по столбцу с одинаковым именем. Если таких несколько, то будет обьединять по всем

select country\_name, country\_id, region\_name from regions

natural join

countries;

1. **JOIN USING**

select first\_name, last\_name, salary, department\_name, e.manager\_id, department\_id

from employees e JOIN departments d *USING* (department\_id);

1. **NATURAL JOIN with ON**

select first\_name, last\_name, start\_date, end\_date, jh.job\_id

from employees

join job\_history jh ON (employees.employee\_id = jh.employee\_id);

SELECT first\_name, last\_name, start\_date, end\_date, jh.job\_id from employees e JOIN job\_history jh ON (e.employee\_id = jh.employee\_id AND e.department\_id = jh.department\_id);

Обьединение нескольких таблиц

select first\_name,last\_name, jh.job\_id, department\_name, start\_date, end\_date

FROM employees emp JOIN job\_history jh ON (jh.employee\_id=emp.employee\_id)

JOIN departments dep ON (jh.department\_id=dep.department\_id);

**4. NONEQUIJOIN** – обьединение таблиц с помощью операторов неравенств

select first\_name, salary, min\_salary, max\_salary

from employees e join jobs j ON(e.job\_id = j.job\_id and salary\*2 < max\_salary);

**5. SELF JOIN – обьединение таблицы с собой**