**ROWNUM and ROWID diff?**

* ROWNUM describes a row wise number that is assigned to each row of a result set. It starts with 1 and increments by 1.

QUERY: select first\_name,rownum from employees;

OUTPUT:

|  |  |
| --- | --- |
| **FIRST\_NAME** | **ROWNUM** |
| Ellen | 1 |
| Sundar | 2 |
| Mozhe | 3 |
| David | 4 |

* ROWID is a 16-digit unique alpha character assigned to every row number. It is used to access rows directly in the database.

QUERY: select first\_name,rowed from employees;

OUTPUT:

|  |  |
| --- | --- |
| **FIRST\_NAME** | **ROWID** |
| Ellen | AAAC9EAAEAAAABXABK |
| Sundar | AAAC9EAAEAAAABXABC |
| Mozhe | AAAC9EAAEAAAABXAAe |
| David | AAAC9EAAEAAAABXAAF |

**What is synonym and purpose?**

* It is a database object
* It is used to provide shorter or more descriptive name for the database object
* Synonym can be created for only whole table
* It is a physical representation of the table

SYNTAX: create synonym synonym\_name for object\_name;

create synonym cs for customer\_service;

* Once there is a data change in synonym name of the table it will also affects the changes in base table too.
* Synonym works in all the database object like sequence,table,view etc..
* Also, we can create synonym name for a synonym name.

Eg: create synonym c for cs;

**What is sequence and purpose?**

* It is a database object
* It is used in primary key constraints
* It works only for number data type

SYNTAX: create sequence sequence\_name

Starts with start\_value

Increment by increment\_value

Maxvalue max\_value;

* It is used to generate sequence of numbers and to generate unique values.

**What is view - why? – functionality?**

* View is a database object
* It is used to store query in the database
* It acts as a virtual table

TYPES OF VIEW:

* SIMPLE VIEW – Only one table involves by getting the data and DML is possible
* COMPLEX VIEW – if the query consists of JOINS, SUBQUERY, SETOPERATOR, PSEUDO COLUMNS, DISTINCT/UNIQUE, FUNCTIONS it is complex view, DML is not possible

ADVANTAGES OF VIEW:

* To restrict data access-it will not allow access to see other data
* Complex query into simple- more lines of query by creating a simple view table to run.

SYNTAX: create or replace view view\_name

as

select columns from tables where condition;

For eg: we have a query\_ select hire\_date from employees;

Instead of running the query we can able to shorten this by putting into view.

Create or replace view v1

as

select hire\_date from employees;

select \* from v1; #it will display the same output as previous query.

**delete and truncate difference?**

Delete:

* Delete is one of the DML operation
* It is used to delete rows from the table

SYNTAX: DELETE FROM table\_name WHERE condition;

DELETE FROM employees WHERE department\_id=60;

#it deletes the row that contains department\_id=60.

* We can delete single row or multiple rows based on condition given.

Truncate:

* Truncate is one of the DDL operation.
* Truncate is used to delete the data in the table.

SYNTAX: TRUNCATE table table\_name;

TRUNCATE table customers;

#it entirely deletes the data in customer table.

**Inline view?**

* The query that is place after ‘from’ clause.

SYNTAX: select columns from (SELECT columns FROM table\_name[WHERE condition]) where condition;

Query: we have a query select first\_name,hire\_date,salary from employees;

It will display output as three columns.

Instead we have a query:

Select first\_name,hire\_date from (select first\_name,hire\_date,salary from employees) where hire\_date>2014;

We can fetch data from inline query columns.

**Who is getting the maximum salary ?**

select first\_name,salary from employees where salary=(Select max(salary) from employees);

OUTPUT:

|  |  |
| --- | --- |
| **FIRST\_NAME** | **SALARY** |
| Steven | 24000 |

**How to get no of records in a table?**

select count(\*) as total\_count from employees;

OUTPUT:

|  |
| --- |
| **TOTAL\_COUNT** |
| 107 |

**Listagg?**

* This is used to concatenate values from multiple rows into a single string.It is particularly useful when you want to combine values from a column across multiple rows.

QUERY: Select listagg(first\_name,’-’)within group(order by first\_name asc) from employees;

OUTPUT: Naveen-pravin-kavin

**How to modify a column datatype in a table which contains millions of records?**

employees last\_name column has char(30)

modifying column datatype

alter table employees modify last\_name varchar(40);