Oracle Tables

WHAT IS ORACLE DATABASE TABLE?

-Tables are the basic unit of data storage in an Oracle Database. Data is stored in rows and columns.

-A table holds all the necessary about something in the real world

-A table contains a set of column. A column represents one kind of data in the table for example, salary column in EMP table will have the salaries

- A row is a collection of column information corresponding to a single record.

Next we will be talking in detail about Oracle create table statement

CREATING THE ORACLE DATABASE TABLE

To create a table in the database, we must have following information

1) The table name

TABLE NAMING CONVENTIONS -

- The name you choose for a table must follow these standard rules:
- The name must begin with a letter A-Z or a-z
- Can contain numbers and underscores
- Can be in UPPER of lower case
- Can be up to 30 characters in length
- Cannot use the same name of another existing object in your schema
- Must not be an Oracle server and SQL reserved word

2) Column name, column data types, and column sizes.

COLUMN NAMING CONVENTIONS -

- The name you choose for a column must follow these standard rules:
- The name must begin with a letter A-Z or a-z
- Can contain numbers and underscores
- Can be in UPPER of lower case
- Can be up to 30 characters in length
- Cannot use the same name of another existing object in your schema
- Must not be an Oracle server and SQL reserved word

Data types

Character	-CHAR, NCHAR, VARCHAR2 & NVARCHAR2. - The CHAR datatype is a fixed-length alphanumeric string which has a maximum length in bytes. -When creating a CHAR datatype, the database will preserve space for the incoming data and if the data is shorter than maximum size, it will be space-padded to the right -The VARCHAR2 datatype is a variable-length alphanumeric string, which has a maximum length in bytes. It can store up to 4000 bytes.
Number	-NUMBER -The NUMBER datatype stores number with precision and scale. -Numeric datatypes store negative and positive integers fixed-point numbers and floating-point numbers - When a column is defined as NUMBER (6, 2), the range of values can be stored from –9999.99 to 9999.99. Oracle rounds the floating-point numbers.
Date and Time	–DATE, TIMESTAMP (with time zone or local time zone), INTERVAL YEAR TO MONTH & INTERVAL DAY TO

	SECOND. -The DATE data type is used to store date and time information. -This data type has a number of specific functions for manipulating, formatting and viewing its data. - The DATE data type holds storage of seven bytes and has information about century, year, month, day, hours, minutes & seconds. - The NLS_DATE_FORMAT parameter can be changed to control the viewing of the data. The SYSDATE function returns the current date
Large Objects	-BLOB *(binary large object)*, CLOB *(character large object)*, NCLOB & BFILE -Columns of these datatypes can store unstructured data including text, image, video, and spatial data. -The CLOB datatype can store up to eight terabytes of character data using the CHAR database character set. -The BLOB datatype is used to store unstructured binary large objects such as those associated with image and video data where the data is simply a stream of "bit" values. -The BFILE data type value works as a file locator or pointer to file on the server's file system. The maximum

	file size supported is 8TB to 128TB.
Long	Variable length character data up to 2 G
rowid	A 64 base number system representing the unique address of the row in the table

3) Type of table

Ordinary (heap-organized) table	-This is the basic, general purpose type of table. -Its data is stored as an unordered collection (heap)
Clustered table	-A clustered table is a table that is part of a cluster. -A cluster is a group of tables that share the same data blocks because they share common columns and are often used together.
Index-organized table	-Unlike an ordinary (heap-organized) table, data for an index-organized table is stored in a B-tree index structure in a primary key sorted manner. -Besides storing the primary key column values of an index-organized table row, each index entry in the B-tree stores the nonkey column values as well.
Partitioned table	-Partitioned tables allow your data to be broken down into smaller, more manageable pieces called partitions,

	or even subpartitions. - Each partition can be managed individually, and can operate independently of the other partitions, thus providing a structure that can be better tuned for availability and performance.
External Table	External tables allow Oracle to query data that is stored outside the database in flat files.
Global temporary table	The data in a global temporary table is private, such that data inserted by a session can only be accessed by that session. The session-specific rows in a global temporary table can be preserved for the whole session, or just for the current transaction

4) Constraints and Rules

You can specify rules for each column of a table. These rules are called integrity constraints. One such example is a not null integrity constraint. This constraint forces the column to contain a value in every row. These rules are enforced placed for each column or set of columns. Whenever the table participates in data action, these rules are validated and raise exception upon violation.

A constraint can be one of the following:

- a column-level constraint

  Column-level constraints refer to a single column in the table and do not specify a column name (except check constraints). They refer to the column that they follow.

- a table-level constraint

  Table-level constraints refer to one or more columns in the table. Table-level constraints specify the names of the columns to which they

apply. Table-level CHECK constraints can refer to 0 or more columns in the table.

The available constraint types are NOT NULL, Primary Key, Unique, Check, and Foreign Key.

Primary Key	A primary key is a column in a table whose values uniquely identify the rows in the table. A primary key value: ▪ Must uniquely identify the row; ▪ cannot have NULL values; Oracle internally creates unique index to prevent duplication in the column values. It can be defined at the column or table level
Check constraint	A **check constraint** requires a value in the database to comply with a specified condition. Check constraint allows to impose a conditional rule on a column, which must be validated before data is inserted into the column. The condition must not contain a sub query or pseudo column CURRVAL NEXTVAL, LEVEL, ROWNUM, or SYSDATE. Oracle allows a single column to have more than one CHECK constraint. In fact, there is no practical limit to the number of CHECK constraints that can be defined for a column. It can be defined at the column or table level
Not Null	It means that a data row must have a value for the column specified as

	NOT NULL. The Oracle server will not allow rows to be stored that violate this constraint. It can only be defined at column level, and not at the table level.
Unique constraints	It means uniqueness for the column. Oracle server will not allow duplicate values in the column having unique constraints. Oracle internally creates unique index to prevent duplication in the column values. But it allows some values to be null. It can be defined at the column or table level
Foreign Key	-A foreign key is a referential constraint between two tables. -A foreign key constraint validates the values an INSERT or UPDATE against the values in another column, either in a different table or another column in the same -A foreign key always defines a parent/child relationship. The "parent" is the column that is referenced in the foreign key and the "child" is the column or columns that contain the foreign key constraint. -Generally, though, a foreign key is a field (or fields) that points to the primary key of another table. -It can be defined at the column or table level

5) Table storage parameter

Tables are stored in Tablespace in the database. If no Tablespace is specified, the table goes in user default Tablespace.

Once we have all the required information, we can move forward with table creation

SYNTAX FOR ORACLE CREATE TABLE STATEMENT

```
CREATE TABLE table_name
(
  col1 datatype [ NULL | NOT NULL ],
  col2 datatype [ NULL | NOT NULL ],
  ...
  col_n datatype [ NULL | NOT NULL ]
) tablespace <tablepace name>;
```

SYNTAX FOR TABLE CREATION WITH PRIMARY KEY. IT CAN BE BOTH DEFINED AT COLUMN LEVEL OR TABLE LEVEL

Table level

```
CREATE TABLE table_name
(
  col1 datatype [ NULL | NOT NULL ],
  col2 datatype [ NULL | NOT NULL ],
  ...
  col_n datatype [ NULL | NOT NULL ]
constraint <name> primary key (col1,col2)
) tablespace <tablepace name>;
```

Column Level

```
CREATE TABLE table_name
```

```
(
  col1 datatype [ NULL | NOT NULL ] constraint <name> primary key
,
  col2 datatype [ NULL | NOT NULL ],
  ...
  col_n datatype [ NULL | NOT NULL ]
)  tablespace <tablepace name>;
```

Example:

```
CREATE TABLE SCOTT.TEST
        (
        Created_by date,
        SOURCE CHAR(10),
        REQUEST_ID CHAR(64) NOT NULL CONSTRAINT TEST_PK PRIMARY
KEY,
        COMMENTS VARCHAR(3000)
   );


CREATE TABLE SCOTT.TEST1
        (
        INV_ID CHAR(7) NOT NULL,
        ITEM_ID CHAR(7) NOT NULL,
        CREATED  date,
        WHO CHAR(7),
        CONSTRAINT TEST1_PK PRIMARY KEY (INV_ID,ITEM_ID)
        );
```

Explanation for table TEST1

1	*The first column is called INV_ID which is created as a char datatype (maximum 7 digits in length) and cannot contain null values*
2	*The second column is called ITEM_ID which is created as a char datatype (maximum 7 digits in length) and cannot contain null values*
3	*The third column is called CREATED which is a date datatype and also can contain null values.*
4	*The fourth column is called WHO which is a cahe datatype and also can contain null values.*
5	*Table level primary key constraint TEST1_PK is defined on the composite key (INV_ID, ITEM_ID)*

SYNTAX FOR TABLE CREATION WITH FOREIGN KEY. IT CAN BE BOTH DEFINED AT COLUMN LEVEL OR TABLE LEVEL

```
CREATE TABLE table_name
(
  col1 datatype [ NULL | NOT NULL ],
  col2 datatype [ NULL | NOT NULL ],
  ...
  col_n datatype [ NULL | NOT NULL ]
constraint <name> FOREIGN KEY (col1,col2) REFERENCES table(col1,col2)
)  tablespace <tablepace name>;


CREATE TABLE table_name
(
  col1 datatype [ NULL | NOT NULL ] constraint <name> primary key
,
  col2 datatype [ NULL | NOT NULL ],
  ...
  col_n datatype [ NULL | NOT NULL ]
)  tablespace <tablepace name>;
```

```
CREATE TABLE dept
( dept_id number(10) NOT NULL,
  dept_name varchar2(50) NOT NULL,
  CONSTRAINT dept_pk PRIMARY KEY (dep_id)
);

CREATE TABLE emp
( emp_no number(10) NOT NULL,
  emp_name varchar2(50) NOT NULL,
  dept_id number(10),
  sal number(6),
  CONSTRAINT emp_pk PRIMARY KEY (emp_no),
  CONSTRAINT dept_fk
    FOREIGN KEY (dept_id)
    REFERENCES dept(dept_id)
```

Explanation for table EMP

1	The first column is called EMP_NO which is created as a number and cannot contain null values
2	The second column is called EMP_NAME which is created as varchar2(50) and cannot contain null values
3	The third column is called DEPT_ID which is a date number.
4	The fourth column is called SAL which is a number datatype and also can contain null values.
5	Table level primary key constraint EMP_PK is defined on the key (EMP_NO)
6	Table level Foreign Key constraints dept_fk which references dept table dept_id

PRIVILEGE REQUIRED TO CREATE TABLE

-You must have the create table system privilege in order to create a new table in your schema,

- You must have the create any table system privilege in order to create a table in another user's schema, additionally, the owner of the table must have a quota for the tablespace that contains the table, or the UNLIMITED TABLESPACE system privilege

OTHER CHARACTERISTICS ASSOCIATED WITH TABLE

Cache/nocache	Use the CACHE clauses to indicate how Oracle Database should store blocks in the buffer cache. If you don't specify anything in create table command, it is by default nocache
DEFAULT	The value inserted into the column if the insert or update would leave the column value NULL. <DEFAULT <value> | NULL>
PARALLEL	Specify PARALLEL if you want Oracle to select a degree of parallelism equal to the number of CPUs available on all participating instances times the value of the PARALLEL_THREADS_PER_CPU initialization parameter.
Compress	This is used by oracle to store the table in compressed format. This is available from 11g onwards

Comment a table or column	You can comment on table using the command COMMENT ON TABLE <table_name> IS '<comment>';

HOW TO MODIFY THE ORACLE DATABASE TABLE?

-You alter a table using the ALTER TABLE statement.

- The table must be contained in your schema to alter a table, or you should either have ALTER object privilege for the table or the ALTER ANY TABLE system privilege.

-If a view, materialized view, trigger, function-based index, check constraint, function, procedure of package depends on a base table, the alteration of the base table or its columns can affect the dependent object for example plsql objects become invalid if the dependent table object is changed and you have to make them valid again

Modify physical characteristics (INITRANS or storage parameters)	We can modify the storage parameter of the table using Alter table statement. We can modify inittrans like this ALTER TABLE TABLE_NAME INITRANS 10;
Moving the table to new segments or tablespace	We can move the non partition table or partition of the table to new segment or new tablespace if required Alter table table_name move tablespace <tablespace name> We can even use the move command to change any storage parameter of

	the tables which are not modified using alter table command
Alter Table Change Data Type	We can change the datatype of any column using alter table modify command ALTER TABLE <table_name> MODIFY (<column_name> <new_data_type>);
Add a new column	We can add new column to the table definition Using alter table add command ALTER TABLE <table_name> ADD (<New column_name> <new_data_type>); -if a new column is added to a table, the column is initially NULL unless you specify the DEFAULT clause. When you specify a default value, the database updates each row in the new column with the values specified.
Rename existing column in the table	Oracle allows you rename existing columns in a table. Use the RENAME COLUMN clause of the ALTER TABLE statement to rename a column.
Drop the column in the table	Oracle allows you to drop the column in the table using the command Alter table <table_name> drop column;

	You can drop multiple col in one command also
Set A Column Unused and Drop the unused column	if you are concerned about the resource consumed in dropping the column then you can use the ALTER TABLE...SET UNUSED statement. This statement marks one or more columns as unused, but does not actually remove the target column data or restore the disk space occupied by these columns. - A column that is marked as unused is not displayed in queries or data dictionary views, and its name is removed so that a new column can reuse that name. All constraints, indexes, and statistics defined on the column are also removed. Example ALTER TABLE <table_name> SET UNUSED COLUMN <column_name>; We can drop the unused column later on when the resource is lower using the command ALTER TABLE <table_name> Drop UNUSED COLUMN ;
Add, modify or drop integrity constraints associated with the table or you can enable/disable the constraints also	Add constraints is done using alter table add constraints ALTER TABLE EMP ADD CONSTRAINT EMP_FK FOREIGN KEY (DEPT_NO)

	REFERENCES DEPT(DEPT_NO); Dropping Constraints – is done using the ALTER TABLE DROP CONSTRAINT <constraint_name> command; Enabling/Disabling Constraints – Constraints can be created in DISABLE/ENABLE mode or can be disabled or enabled using the ALTER TABLE ENABLE/DISABLE CONSTRAINT <constraint_name> command;
Rename table name	Oracle allows you to rename the table name also Rename <table name> to <new table name>;
Alteration of table cache/nocache, Compression, parallelism	*Oracle allows* Alteration of table cache/nocache, Compression, parallelism

CREATION OF TABLE USING SUB QUERY

-A table can be created from an existing table in the database using a sub query option.

- The table is created with specified column names and rows retrieved by the select statement are inserted into the table

-if the table column specification is not given, table is created with the same column as given in sub query

The below CTAS script creates a new table FND_BACKUP. All the data in FND table is inserted into FND_BACKUP table

```
CREATE TABLE FND_BACKUP
AS
SELECT * FROM FND
```

-Data can also be copied based on conditions or we can put rownum <1 condition to just get the table structure

-The column data type definitions including the explicitly imposed NOT NULL constraints are copied into the new table.

CREATING A TEMPORARY TABLE

-Oracle allows us to create a temporary table.

-The definition of a temporary table is visible to all sessions, but the data in a temporary table is visible only to the session that inserts the data into the table.

-We have to use the CREATE GLOBAL TEMPORARY TABLE statement to create a temporary table.

-The ON COMMIT clause indicates if the data in the table is transaction-specific (the default) or session-specific

DELETE ROWS	This creates a temporary table that is transaction specific. A session becomes bound to the temporary table with a transaction first insert into the table. The binding goes away at the end of the transaction. The database truncates the table (delete all rows) after each commit.
PRESERVE ROWS	This creates a temporary table that is session specific. A session gets bound to the temporary table with the first insert into the table in the session. This binding goes away at the end of the session or by issuing a TRUNCATE of the table in the session. The database truncates the

	table when you terminate the session.

Temporary tables are useful in applications where a result set is to be buffered, perhaps because it is constructed by running multiple DML operations

```
CREATE GLOBAL TEMPORARY TABLE GL_DATA_TEMP
        (startdate DATE,
         enddate DATE,
         gl_id CHAR(20))
      ON COMMIT DELETE ROWS;
```

Indexes can be created on temporary tables. They are also temporary and the data in the index has the same session or transaction scope as the data in the underlying table.

If the TRUNCATE statement is issued against a temporary table, only the session specific data is truncated. There is no effect on the data of other sessions.

If you rollback a transaction, the data you entered is lost, although the table definition persists.

Data in temporary tables is stored in temp segments in the temp tablespace which does not generate any redo so operation using global temporary table are relatively faster. But undo is still generated in undo tablespace which has redo logging. So redo operation is not totally eliminated in Global temporary tables but they are relatively lower

Data in temporary tables is automatically deleted at the end of the database session, even if it ends abnormally.

Views can be created against temporary tables and combinations of temporary and permanent tables. They can be having triggers associated with them

With 12.1(12c database) Oracle release, concept of temporary undo has been introduced which allows the undo segments for global temporary tables to be stored in the temporary tablespace. This allows global temporary tables to be used in physical standby databases and read-only databases, as well as removing the need to create redo.

```
ALTER SYSTEM SET TEMP_UNDO_ENABLED = TRUE;
ALTER SYSTEM SET TEMP_UNDO_ENABLED = FALSE;
```

DROP TABLE STATEMENT

-The DROP TABLE command is used to remove a table from the database.

-The dropped table and its data remain no longer available for selection. Dropping a table drops the index and triggers associated with it.

- Views, synonym are not dropped but they become invalid

-Dropped table can be recovered using FLASHBACK utility, if available in recyclebin. This functionality is available from 10g onwards

-Only the creator of table can drop the table or the user with drop any table privilege can drop the table

Syntax for Drop table

DROP TABLE [TABLE NAME] [PURGE]

The below statement will drop the table and place it into the recyclebin.

DROP TABLE TEST;

The below statement will drop the table and flush it out from the recyclebin also.

DROP TABLE TEST PURGE;

TRUNCATE TABLE STATEMENT

- We can use truncate table command to delete all the rows in the table and it releases all the storage space allocated to the table. It resets the high water mark of the table

- This command cannot be rollback

- You should be owner of the table to execute the operation

-Truncating the table is a DDL command and it does not fire the on delete triggers

-We can delete all the rows in the table by delete command also but it does not release the storage space allocated to the table

DATA DICTIONARY TABLES AND VIEWS

All the table and column information are stored in SYS.TAB$ and SYS.COL$ tables. Oracle has provided data dictionary views to get the information about table and columns

There are three categories of views

USER_%	This view contain information of the objects owned by the user only Example USER_TABLES, USER_TAB_COLS
ALL-%	This view contains information of the objects which the user can access in the database. Example

	ALL_TABLES, ALL_TAB_COLS
DBA_%	This view contains information of the all objects in the system and these are restricted views which are accessible to the user who have DBA role Example DBA_TABLES, DBA_TAB_COLS

	DBA_% views about table information	**ALL_% views about table information**	**USER_% views about table information**
View about column comments	dba_col_comments	all_col_comments	user_col_comments
View about external tables	dba_external_tables	all_external_tables	user_external_tables
View about external tables location	dba_external_locations	all_external_locations	user_external_locations
	dba_partial_drop_tabs	all_partial_drop_tabs	user_partial_drop_tabs
View about table information	dba_tables	all_tables	user_tables
View about table column	dba_tab_cols	all_tab_cols	user_tab_cols
	dba_tab_columns	all_tab_columns	user_tab_columns

	dba_tab_col_statistics	all_tab_col_statistics	user_tab_col_statistics
	dba_tab_comments	all_tab_comments	user_tab_comments
	dba_tab_histograms	all_tab_histograms	user_tab_histograms
View about table monitoring	dba_tab_modifications	all_tab_modifications	user_tab_modifications
View table privilege	dba_tab_privs	all_tab_privs	user_tab_privs
	dba_tab_statistics	all_tab_statistics	user_tab_statistics
	dba_tab_stats_history	all_tab_stats_history	user_tab_stats_history
View about unused column in tables	dba_unused_col_tabs	all_unused_col_tabs	dba_unused_col_tabs

To list all tables owned by the current user, type:

select tablespace_name, table_name from user_tables;

To list all tables in a database:

select tablespace_name, table_name from dba_tables;

To list all tables accessible to the current user, type:

select tablespace_name, table_name from all_tables

To describe the table in sqlplus

desc <table_name>

HOW TO DETERMINE TABLE SIZE?

```
select
  owner as "Schema"
  , segment_name as "Object Name"
  , segment_type as "Object Type"
  , round(bytes/1024/1024,2) as "Object Size (Mb)"
  , tablespace_name as "Tablespace"
from dba_segments
where segment_name='<table_name>';
```