SQL

Components of SQL

DDL (Data Definition Language) –

It is a set of SQL commands used to create, modify, and delete database structures but not data.
Ex: CREATE, ALTER, DROP, TRUNCATE, COMMENT, GRANT, REVOKE

DML (Data Manipulation Language) –

It is the area of SQL that allows changing data within the database.
Ex: INSERT, UPDATE, DELETE, CALL, LOCK

DCL (Data Control Language) –

It is the component of SQL statement that control access to data and to the database.
Ex: COMMIT, SAVEPOINT, ROLLBACK, GRANT, REVOKE

Oracle Basic Data Types:

CHAR (size):This data type is used to store character strings of fixed length. The maximum no. of characters this data type can hold is 255 characters

VARCHAR (size)/ VARCHAR2 (size): This data type is used to store variable length alphanumeric data. It can hold 1 to 255 characters.

DATE : This data type is used to represent date and time. The standard format is DD-MM-YY.

NUMBER(P,S):This data type is used to store fixed or floating point nos.

LONG: This data type is used to store variable length character strings containing upto 2GB.

Create Table:

CREATE TABLE <TableName>
(<ColumnName1> <Data Type (size)>,
<ColumnName2> <Data Type (size)>);
Ex: CREATE TABLE STUDENT
(ROLL_NO VARCHAR2(10), NAME CHAR(25),
BRANCH VARCHAR2(20), PERCENT NUMBER(4,2));

DESC COMMAND

This command is used to view the structure or schema of any table.
Syntax:
DESC <Table Name>;
Ex: DESC STUDENT;

RENAMING TABLES

RENAME command is used to change the old name of any table to a new one.
Syntax: RENAME <OldTableName> TO <NewTableName>;
Ex: RENAME STUDENT TO STU;

Adding New Columns:

This command is used to add anew column at the end of the structure of an existing table.
Syntax: ALTER TABLE <TableName> ADD(<NewColumnName> <Datatype>(<size>),
<NewColumnName> <Datatype>(<size>),…….);
Ex: ALTER TABLE STUDENT ADD(AGE NUMBER(2));

Dropping a Column from the Table
This is used for removing a column of any existing table.
Syntax: ALTER TABLE <TableName> DROP COLUMN <ColumnName>;
Ex: ALTER TABLE STUDENT DROP COLUMN AGE;

Modifying Existing Columns
This is used to modify/change the size of any column in a table.
Syntax: ALTER TABLE <TableName> MODIFY (<ColumnName> <NewDataType> (<NewSize>));
Ex: ALTER TABLE STUDENT MODIFY(ROLL_NO VARCHAR2(15);

DESTROYING TABLES
DROP TABLE command is used to delete/discard any table.
Syntax: DROP <TableName>;

TRUNCATING TABLES
TRUNCATE TABLE command deletes all the rows of any table.
Syntax: TRUNCATE TABLE <STU>;

PRIMARY KEY
The primary key of a relational table uniquely identifies each record in the table. Primary keys may consist of a single attribute or multiple attributes in combination.
Syntax:
CREATE TABLE <TableName>(<ColumnName> <DataType>(<Size>) PRIMARY KEY,…….);

Ex:

 CREATE TABLE Customer
(SID integer PRIMARY KEY,
Last_Name varchar(30),
First_Name varchar(30));

 ALTER TABLE Customer ADD PRIMARY KEY (SID);

FOREIGN KEY
A foreign key is a field (or fields) that points to the primary key of another table. The purpose of the foreign key is to ensure referential integrity of the data. In other words, only values that are supposed to appear in the database are permitted.
Syntax:
<ColumnName> <DataType>(<Size>)
REFERENCES <TableName>[(<ColumnName>)]…
Ex:

 CREATE TABLEORDERS
(Order_ID integer primary key,
Order_Date date,
Customer_SID integer references CUSTOMER(SID),
Amount double);

 ALTER TABLE ORDERSADD
(CONSTRAINT fk_orders1) FOREIGN KEY (customer_sid) REFERENCES CUSTOMER(SID);

UNIQUE Key:
Syntax:
<ColumnName> <DataType>(<Size>) UNIQUE

CHECK CONSTRAINT
A check constraint allows you to specify a condition on each row in a table.
Note:
 A check constraint can NOT be defined on a VIEW.
 The check constraint defined on a table must refer to only columns in thattable. It can not refer to columns in other tables.
 A check constraint can NOT include a SUBQUERY.
Ex:

CREATE TABLE suppliers
(supplier_idnumeric(4),
supplier_namevarchar2(50),
CONSTRAINT check_supplier_id CHECK (supplier_id BETWEEN 100 and 9999));

ALTER TABLE suppliers add CONSTRAINT check_supplier_name CHECK (supplier_name IN ('IBM', 'Microsoft', 'Nvidia'));

NOT NULL CONSTRAINT:
CREATE TABLE newitems (newitem_num INTEGER, menucode CHAR(3) NOT NULL, promotype INTEGER, descrip CHAR(20))

VIEWS:
Syntax:
CREATE VIEW <view_name> AS
SELECT columnname, columnname
FROM <table_name>
Where columnname = expression list;
Group By grouping criteria
Having predicate
Note: The Order By clause cannot be used while creating a view.

Selecting a data set from the view:
Once a view has been created, it can be queried exactly like a base table.
Syntax:
Select columnname, columnname
From viewname;

Updating views;
Views can also be used for data manipulation(i.e. the user can perform the Insert, Update and Delete operations).
Views on which data manipulation can be done are called Updatable Views. When you give an updatable view name in the Update, Insert or Delete SQL statement, modifications to data will be passed to the underlying table.

A Unique Index
Creates a unique index on a table. A unique index means that two rows cannot have the same index value.
CREATE UNIQUE INDEX index_name
ON table_name (column_name)
The "column_name" specifies the column you want indexed.

A Simple Index
Creates a simple index on a table. When the UNIQUE keyword is omitted, duplicate values are allowed.
CREATE INDEX index_name
ON table_name (column_name)

INSERT COMMAND
This command is used to enter (input) data into the created table.
Syntax:
INSERT INTO <TableName> (<ColumnName1>,<ColumnName2>)
VALUES(<Expression1>,<Expression2>);
Ex: INSERT INTO STUDENT(ROLL_NO,NAME,BRANCH,PERCENT)
VALUES(_CS05111‘, _AAA‘, _CO.SC‘, 84.45);

VIEWING DATA FROM THE TABLES:
All Rows & Columns of a Table-
SELECT * FROM <Table Name>;

ELIMINATING DUPLICATE ROWS WHEN USING A SELECT STATEMENT
Syntax:
SELECT DISTINCT <Column Name1>, <Column Name2>
FROM <Table Name>;
Ex: SELECT DISTINCT NAME, PERCENT FROM STUDENT;
Syntax:
SELECT DISTINCT * FROM <TableName>;
Ex: SELECT DISTINCT * FROM STUDENT;

CREATING A TABLE FROM A TABLE
Using AS SELECT clause in the CREATE TABLE command one can create a new table from any existing table.
Syntax:
CREATE TABLE <TableName> (<ColumnName>, <ColumnName>) AS SELECT
<ColumnName> , <ColumnName> FROM <TableName>;

SORTING DATA IN A TABLE
The rows retrieved from the table will be sorted in either **ascending** or **descending** order.
Syntax:
SELECT * FROM <TableName> ORDER BY <ColumnName1>,<ColumnName2>
<[SORT ORDER]>;
Ex:
i) SELECT * FROM STUDENT ORDER BY ROLL_NO, NAME;
ii) SELECT * FROM STUDENT ORDER BY NAME DESC;

UPDATING THE CONTENTS OF A TABLE
The UPDATE command is used to change or modify data values in a table.
Update All Rows
UPDATE <TableName> SET <ColumnName1> = <Expression1>,
<ColumnName2>=<Expression2>;
Ex: UPDATE STUDENT SET BRANCH=_CS‘;