LECTURE-3

STRUCTURED QUERY LANGUAGE
DATA DEFINITION LANGUAGE (DDL)

Delivered By:

Mr. Shubham Shukla

DATA DEFINITION LANGUAGE (DDL):

Three basic commands:

- CREATE
- DROP
- ALTER

CREATE

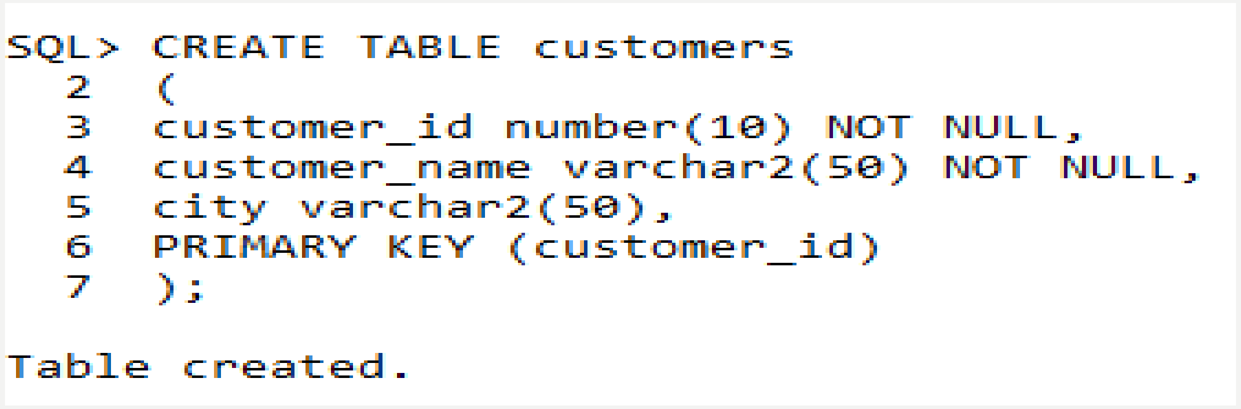
- **CREATE TABLE Statement**

Syntax

```
CREATE TABLE table_name
(
column1 datatype [ NULL | NOT NULL ],
 column2 datatype [ NULL | NOT NULL ],
 ...
column_n datatype [ NULL | NOT NULL ]
);
```

Example:

```
CREATE TABLE customers
(
customer_id number(10) NOT NULL,
customer_name varchar2(50) NOT NULL,
city varchar2(50),
PRIMARY KEY (customer_id)
);
```



```
SQL> CREATE TABLE customers
     (
     customer_id number(10) NOT NULL,
     customer_name varchar2(50) NOT NULL,
     city varchar2(50),
     PRIMARY KEY (customer_id)
     );

Table created.
```

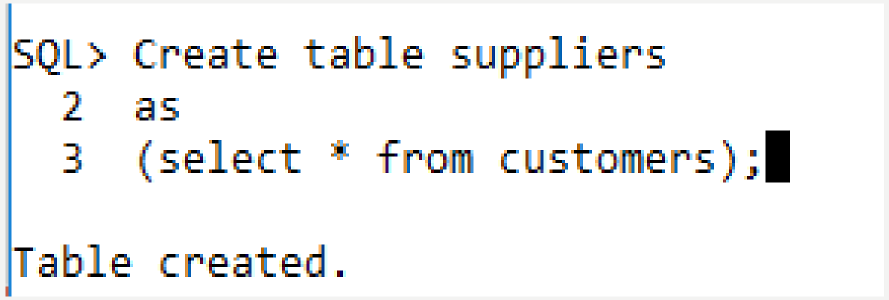
CREATE TABLE AS Statement:

- **Create Table - By Copying all columns from another table**

Syntax

CREATE TABLE new_table

  AS (SELECT * FROM old_table);



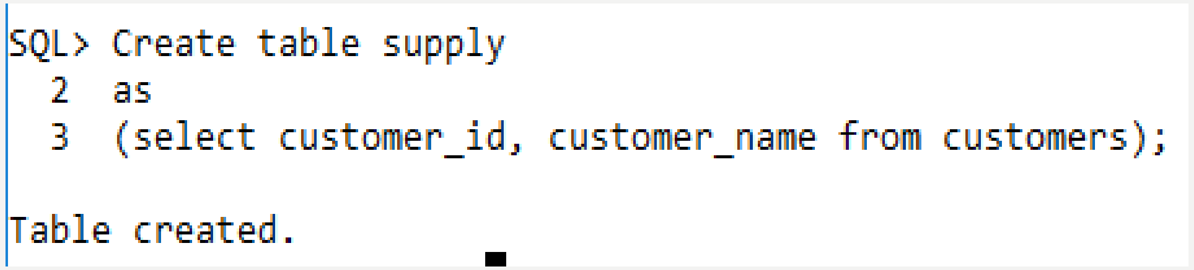
```
SQL> Create table suppliers
  2  as
  3  (select * from customers);

Table created.
```

- **Create Table - By Copying selected columns from another table**

Syntax

CREATE TABLE new_table

  AS (SELECT column_1, column2, ... column_n FROM old_table);



```
SQL> Create table supply
  2  as
  3  (select customer_id, customer_name from customers);

Table created.
```

➢**Create table - By Copying selected columns from multiple tables:**

Syntax

```
CREATE TABLE new_table
  AS (SELECT column_1, column2, ... column_n
      FROM old_table_1, old_table_2, ... old_table_n);
```

Example:

```
CREATE TABLE sup_cust
  AS (SELECT customers.customer_id, customers.city, Supplier1.Supplier_name
      FROM customers, Supplier1
      WHERE customers.customer_id = Supplier1.Supplier_id);
```

ALTER

- **ALTER TABLE Statement**

➢ **Add Single column in table**

Syntax

```
ALTER TABLE table_name
  ADD column_name column-definition;
```

Example:

```
ALTER TABLE customers
  ADD customer_name varchar2(45);
```

➢**Add multiple columns in table**

Syntax

```
ALTER TABLE table_name
  ADD (column_1 column-definition,
       column_2 column-definition,
       ...
       column_n column_definition);
```

For example:

```
ALTER TABLE customers
  ADD (customer_name varchar2(45),
       city varchar2(40));
```

➢**Modify column in table**

Syntax

```
ALTER TABLE table_name
  MODIFY column_name column_type;
```

For example:

```
ALTER TABLE customers
  MODIFY customer_name varchar2(100) not null;
```

- **Modify Multiple columns in table**

Syntax

```
ALTER TABLE table_name
  MODIFY (column_1 column_type,
          column_2 column_type,
          ...
          column_n column_type);
```

For example:

```
ALTER TABLE customers
  MODIFY (customer_name varchar2(100) not null,
          city varchar2(75));
```

➢Drop column in table

Syntax

```
ALTER TABLE table_name
  DROP COLUMN column_name;
```

For example:

```
ALTER TABLE customers
  DROP COLUMN customer_name;
```

➢**Rename column in table**

Syntax

```
ALTER TABLE table_name
  RENAME COLUMN old_name to new_name;
```

For example:

```
ALTER TABLE customers
  RENAME COLUMN customer_name to cname;
```

➢**Rename table**

Syntax

```
ALTER TABLE table_name
  RENAME TO new_table_name;
```

For example:

```
ALTER TABLE customers
  RENAME TO contacts;
```

DROP

- **DROP TABLE Statement**

Syntax

```
DROP TABLE [schema_name].table_name
[ CASCADE CONSTRAINTS ]
[ PURGE ];
```

CASCADE CONSTRAINTS: Optional. If specified, all referential integrity constraints will be dropped as well.

PURGE: Optional. If specified, the table and its dependent objects will be purged from the recycle bin and you will not be able to recover the table. If not specified, the table and its dependent objects are placed in the recycle bin and can be recovered later, if needed.

For example:

```
DROP TABLE customers PURGE;
```

VIEW

- It is a virtual table that does not physically exist. Rather, it is created by a query joining one or more tables.

➢ **Create VIEW**

Syntax

```
CREATE VIEW view_name AS
  SELECT columns
  FROM tables
  [WHERE conditions];
```

VIEW

Example:

```
CREATE VIEW sup_orders AS
  SELECT suppliers.supplier_id, orders.quantity, orders.price
  FROM suppliers
  INNER JOIN orders
  ON suppliers.supplier_id = orders.supplier_id
  WHERE suppliers.supplier_name = 'Microsoft';
```

- **Update VIEW**

Syntax

```
CREATE OR REPLACE VIEW view_name AS
  SELECT columns
  FROM table
  WHERE conditions;
```

Example

```
CREATE or REPLACE VIEW sup_orders AS
  SELECT suppliers.supplier_id, orders.quantity, orders.price
  FROM suppliers
  INNER JOIN orders
  ON suppliers.supplier_id = orders.supplier_id
  WHERE suppliers.supplier_name = 'Apple';
```

➢Drop VIEW

Syntax

```
DROP VIEW view_name;
```

Example

```
DROP VIEW sup_orders;
```

THANK YOU