Enabling DB Concepts

Structured query language (SQL)

DR. MOHAMMED A. MOHAMMED

UNIVERSITY OF SULAIMANI | COLLEGE OF SCIENCE | COMPUTER DEPT.

2024 - 2025

What are DB and DBMS?

- Database: Database is an organized collection of information. It is organized in such a way as to enable easy, optimized storage and use (adding, updating, and searching of data) of large quantities of information.

- DBMS(Database Management System): It is a software package designed to store and manage databases. It enables data to be made available to users for viewing, entry of new data, or updating of existing data, while protecting the rights of those same users.

DBMS

File Database(Offline)

- Microsoft Access
- SQLite
- Other ……

Server Database(Online)

- MySQL Server
- Microsoft SQL Server
- Oracle XE
- Other….

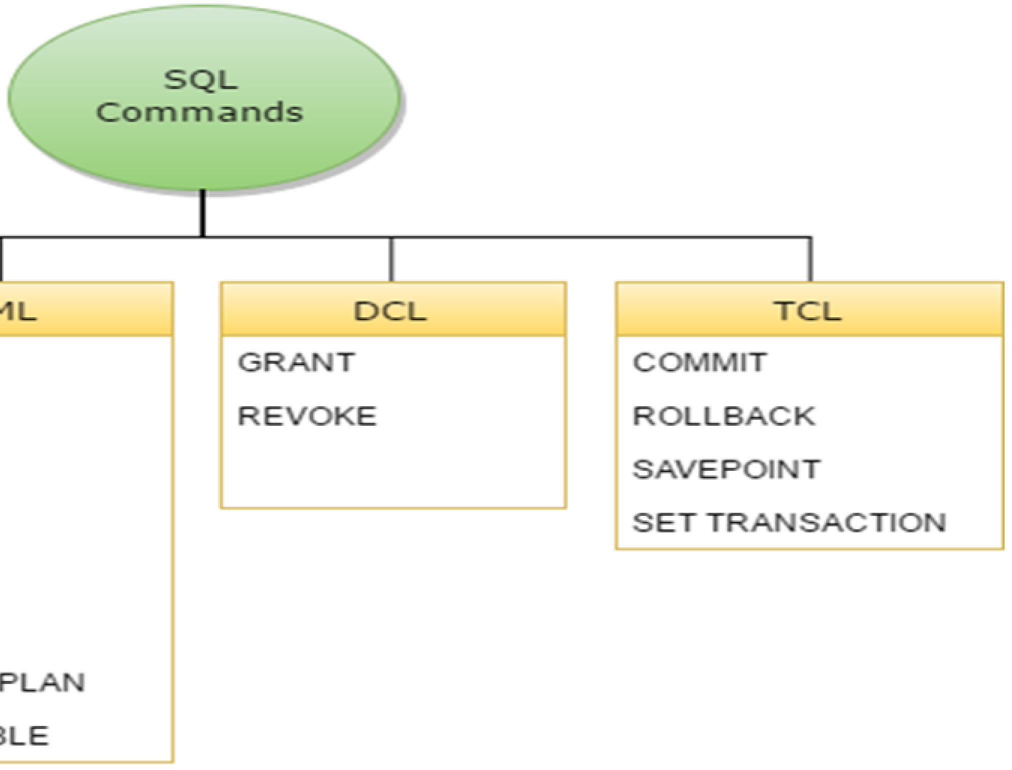
What is SQL?

- SQL stands for Structured Query Language.
- SQL lets you access and manipulate databases.
- SQL is an ANSI (American National Standards Institute) standard.

What Can SQL do?

1. SQL can execute queries against a database.
2. SQL can retrieve data from a database.
3. SQL can insert records in a database.
4. SQL can update records in a database.
5. SQL can delete records from a database.
6. SQL can create new databases.
7. SQL can create new tables in a database.
8. SQL can create stored procedures in a database.
9. SQL can create views in a database.
10. SQL can set permissions on tables, procedures, and views.

Structure Query Language (SQL)



Working on Databases

1 **Create Structure Database**

Data Definition Language(DDL)
it is used to define data **structures.**

SQL Command:
- **Create**
- **Alter**
- **Drop**
- **Truncate**
- **….etc**

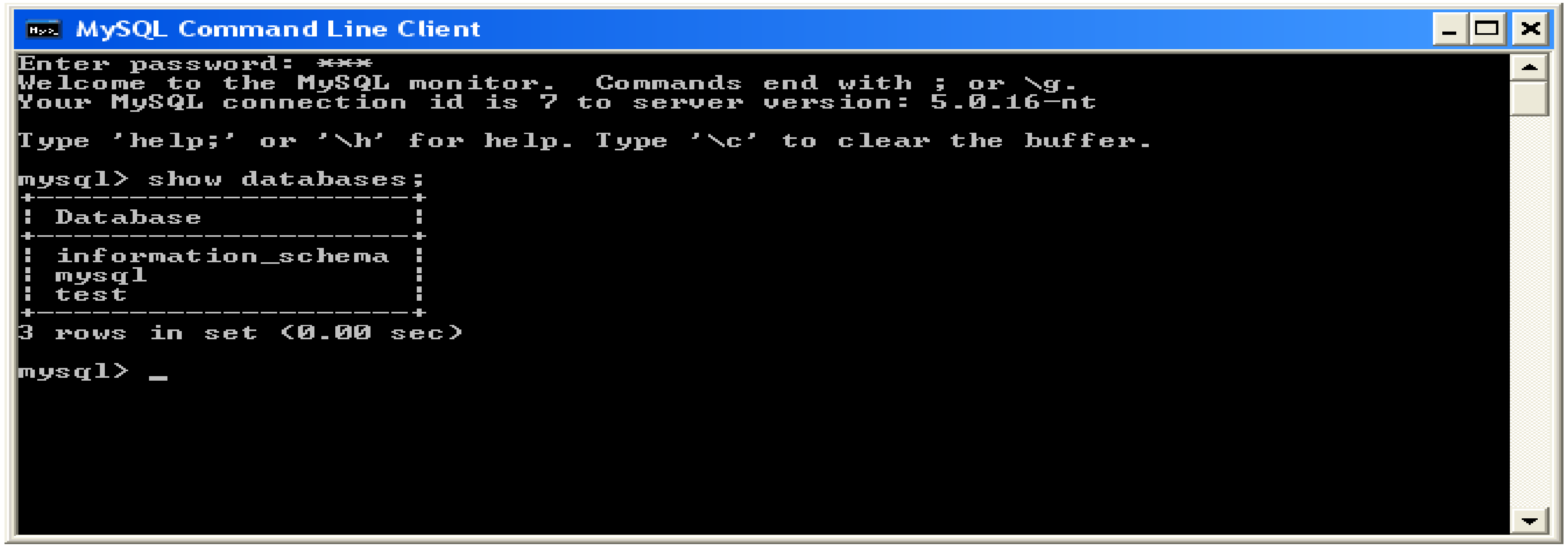
2 **Management Content Database**

Data Manipulation Language(DML)
it is used to manipulate **data itself**

SQL Command:
- **Insert**
- **Update**
- **Delete**
- **..…etc**

At first working on MySQL command line

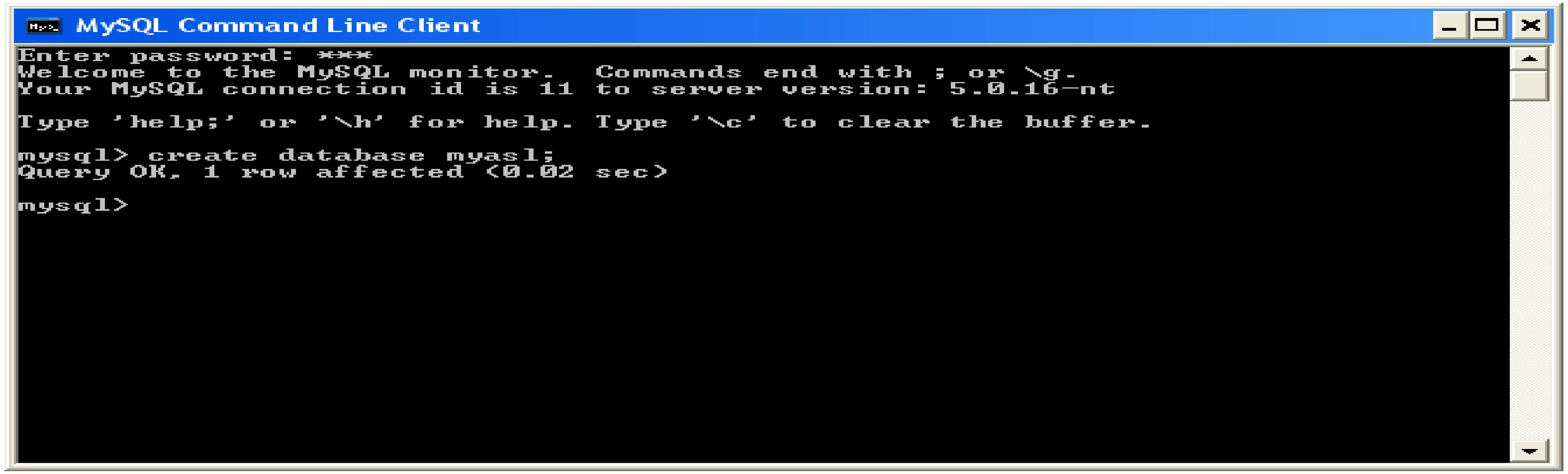
- Show Databases;



```
Enter password: ***
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 7 to server version: 5.0.16-nt

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| test               |
+--------------------+
3 rows in set (0.00 sec)

mysql> _
```

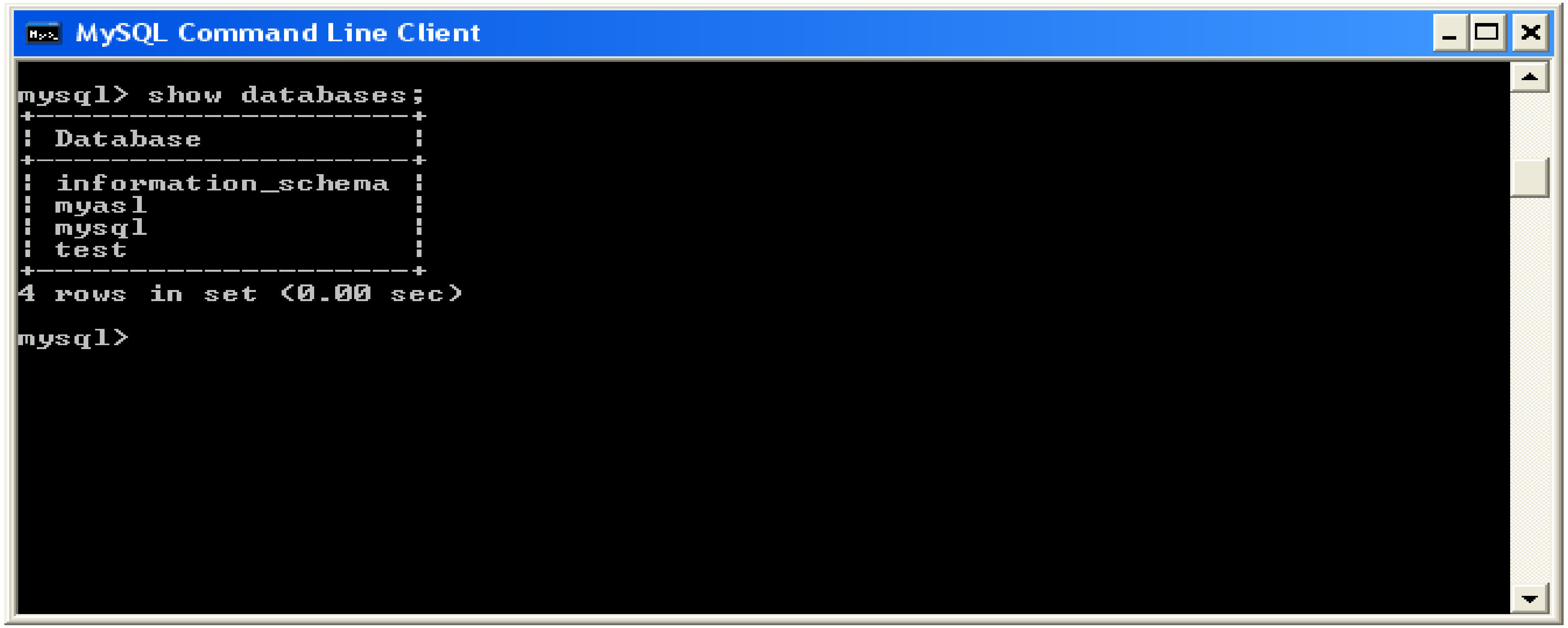
Creating new database

- Create Database **name** ;
- for instance: Create Database **myasl**;



```
Enter password: ***
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 11 to server version: 5.0.16-nt

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> create database myasl;
Query OK, 1 row affected (0.02 sec)

mysql>
```

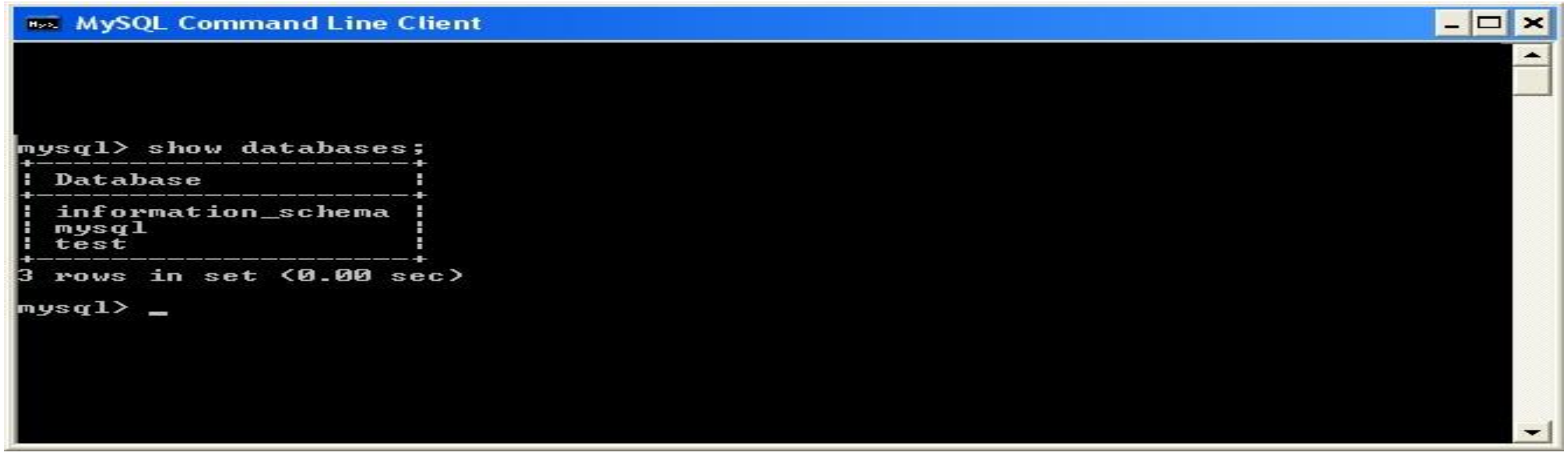
The new database



```
mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| myasl              |
| mysql              |
| test               |
+--------------------+
4 rows in set (0.00 sec)

mysql>
```

Deleting a database

- drop database **name**;
- for instance: drop database **myasl**;



```
mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| test               |
+--------------------+
3 rows in set (0.00 sec)

mysql> _
```

Using databases

- mysql> use mysql;

Database changed

- mysql>

```
CREATE TABLE TestTable(ID INT NOT NULL PRIMARY KEY AUTO_INCREMENT,
   First_Name VAR CHAR(20) NOT NULL, Last_Name VARCHAR(30) NOT NULL, Age
   INT,Gender ENUM('F','M') DE FAULT 'F');
```

- Query OK, 0 rows affected (0.08 sec)

Now working on Oracle Database 10g Express Edition

- For information about how to install **Oracle Database 10g Express Edition** please look at **Installing - Oracle 10g XE**.

Creating Table (Syntax)

```
CREATE TABLE table_name
(
column_name1 data_type(size) Constraint,
column_name2 data_type(size) Constraint,
column_name3 data_type(size) Constraint,
…
);
```

Creating Table (Example)

```
CREATE TABLE  tester
    (  ID NUMBER(2,0) PRIMARY KEY,
 NAME NVARCHAR2(1000),
 PRICE NUMBER(2,0)
   )
```

Modifying table

- Alter table **tester**
- add age number(3,0)

- ALTER TABLE **tester**
- MODIFY age NVARCHAR2(100)

- ALTER TABLE **tester**
- DROP COLUMN age

Oracle Data Types: String data type

- **CHAR**
  CHAR should be used for storing **fix length character strings**. If this type is used to store variable length strings, it will waste lots of **disk space**.

-
  ```
  CREATE TABLE test(name CHAR(10));
  ```

  VARCHAR
  Currently **VARCHAR behaves exactly the same as VARCHAR2**. However, this type should not be used as it is reserved for future usage.

  ```
  CREATE TABLE test(name VARCHAR(10));
  ```

Oracle Data Types: String data type

VARCHAR2: is used to store variable length character strings. **The string value's length will be stored on**

```
CREATE TABLE test(name VARCHAR2(10));
```

Note : **NCHAR** AND **NVARCHAR** are used for Unicode

Oracle Data Types: Integer data type

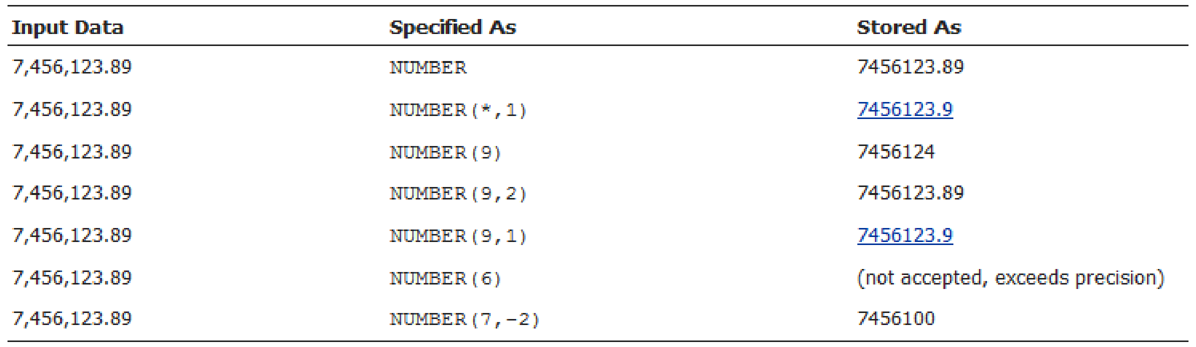
- Numerical data types

```
NUMBER(Precision , Scale );
DECIMAL(Precision , Scale );
```

- Number ,decimal and integer are the same .

Oracle Data Types: Integer data type (Cont.)

- **Precision** is the total number of digits.
- **Scale** is the number of digits after the decimal point.



Input Data	Specified As	Stored As
7,456,123.89	NUMBER	7456123.89
7,456,123.89	NUMBER(*,1)	7456123.9
7,456,123.89	NUMBER(9)	7456124
7,456,123.89	NUMBER(9,2)	7456123.89
7,456,123.89	NUMBER(9,1)	7456123.9
7,456,123.89	NUMBER(6)	(not accepted, exceeds precision)
7,456,123.89	NUMBER(7,-2)	7456100

- If you specify a negative scale, Oracle Database rounds the actual data to the specified number of places to the left of the decimal point.
- For example, specifying (7,-2) means Oracle Database rounds to the nearest hundredths.

Example

```
create table TEST (A decimal(*,5), B number (*, 5));
 Table created
```

The **DESCRIBE** command allows you to describe objects

```
desc TEST;
```

Oracle Data Types: Date

```
CREATE TABLE TEST (
c_id NUMBER,
 c_dt DATE
);
```

What are Constraints?

- Constraints enforce rules at the table level.
- Constraints prevent the deletion of a table if there are dependencies.
- The following constraint types are valid:
  - **NOT NULL**
  - **UNIQUE**
  - **PRIMARY KEY**
  - **FOREIGN KEY**
  - **CHECK**

Constraint Guidelines

- Name a constraint or the Oracle server generates a **name** by using the **SYS_C*n*** format.
- Create a constraint either:
  - At the same time as the table is created, or
  - After the table has been created
- Define a constraint at the column or table level.
- View a constraint in the data dictionary.

Defining Constraints

```
CREATE TABLE [schema.]table
            (column datatype [DEFAULT expr]
        [column_constraint],
        ...
        [table_constraint][,...]);
```

```
CREATE TABLE employees(
             employee_id  NUMBER(6),
             first_name   VARCHAR2(20),
             ...
             job_id       VARCHAR2(10) NOT NULL,
             CONSTRAINT emp_emp_id_pk
                          PRIMARY KEY (EMPLOYEE_ID));
```

Defining Constraints

- Column constraint level

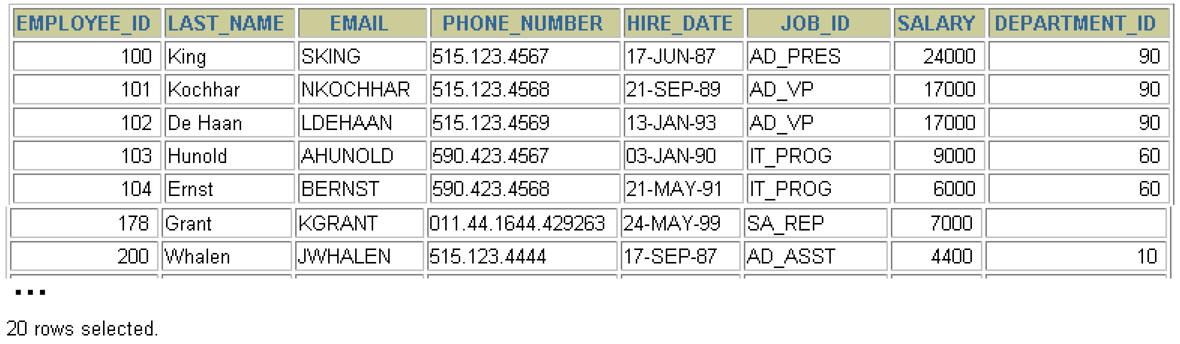
```
column [CONSTRAINT constraint_name] constraint_type,
```

- Table constraint level

```
column,...
  [CONSTRAINT constraint_name] constraint_type
  (column, ...),
```

The NOT NULL Constraint

Ensures that null values are not



EMPLOYEE_ID	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	DEPARTMENT_ID
100	King	SKING	515.123.4567	17-JUN-87	AD_PRES	24000	90
101	Kochhar	NKOCHHAR	515.123.4568	21-SEP-89	AD_VP	17000	90
102	De Haan	LDEHAAN	515.123.4569	13-JAN-93	AD_VP	17000	90
103	Hunold	AHUNOLD	590.423.4567	03-JAN-90	IT_PROG	9000	60
104	Ernst	BERNST	590.423.4568	21-MAY-91	IT_PROG	6000	60
178	Grant	KGRANT	011.44.1644.429263	24-MAY-99	SA_REP	7000	
200	Whalen	JWHALEN	515.123.4444	17-SEP-87	AD_ASST	4400	10

...

20 rows selected.

NOT NULL constraint (No row can contain a null value for this column.)

NOT NULL constraint

Absence of NOT NULL constraint (Any row can contain null for this column.)

The NOT NULL Constraint

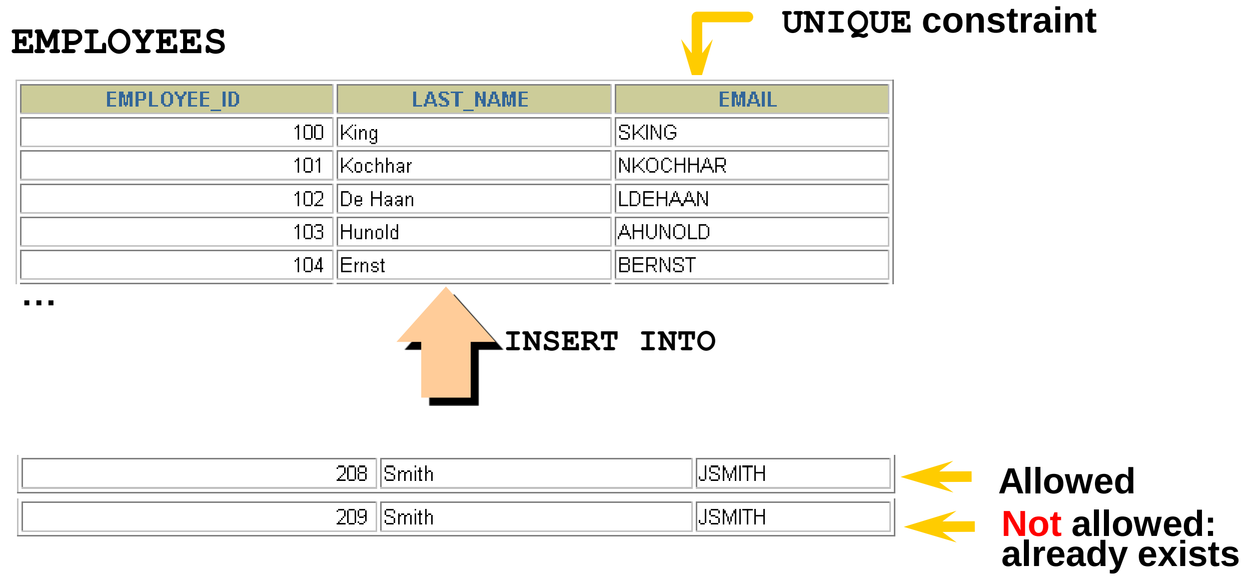
Is defined at the column level:

```
CREATE TABLE employees(
    employee_id    NUMBER(6),
    last_name      VARCHAR2(25) NOT NULL,
    salary         NUMBER(8,2),
    commission_pct NUMBER(2,2),
    hire_date      DATE
                   CONSTRAINT emp_hire_date_nn
                   NOT NULL,
...
```

System named

User named

The UNIQUE Constraint



EMPLOYEE_ID	LAST_NAME	EMAIL
100	King	SKING
101	Kochhar	NKOCHHAR
102	De Haan	LDEHAAN
103	Hunold	AHUNOLD
104	Ernst	BERNST

…

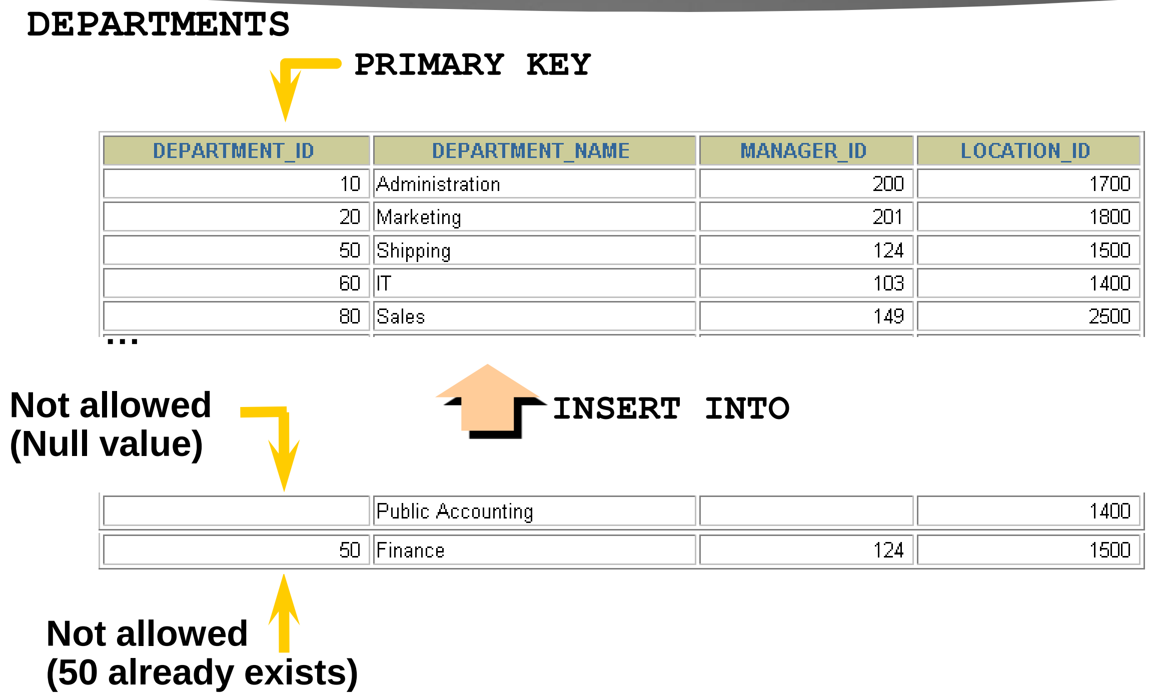
208	Smith	JSMITH
209	Smith	JSMITH

The UNIQUE Constraint

Defined at either the table level or the column level:

```
CREATE TABLE employees(
    employee_id      NUMBER(6),
    last_name        VARCHAR2(25) NOT NULL,
    email            VARCHAR2(25),
    salary           NUMBER(8,2),
    commission_pct   NUMBER(2,2),
    hire_date        DATE NOT NULL,
...
    CONSTRAINT emp_email_uk UNIQUE(email));
```

The PRIMARY KEY Constraint



DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
10	Administration	200	1700
20	Marketing	201	1800
50	Shipping	124	1500
60	IT	103	1400
80	Sales	149	2500

...

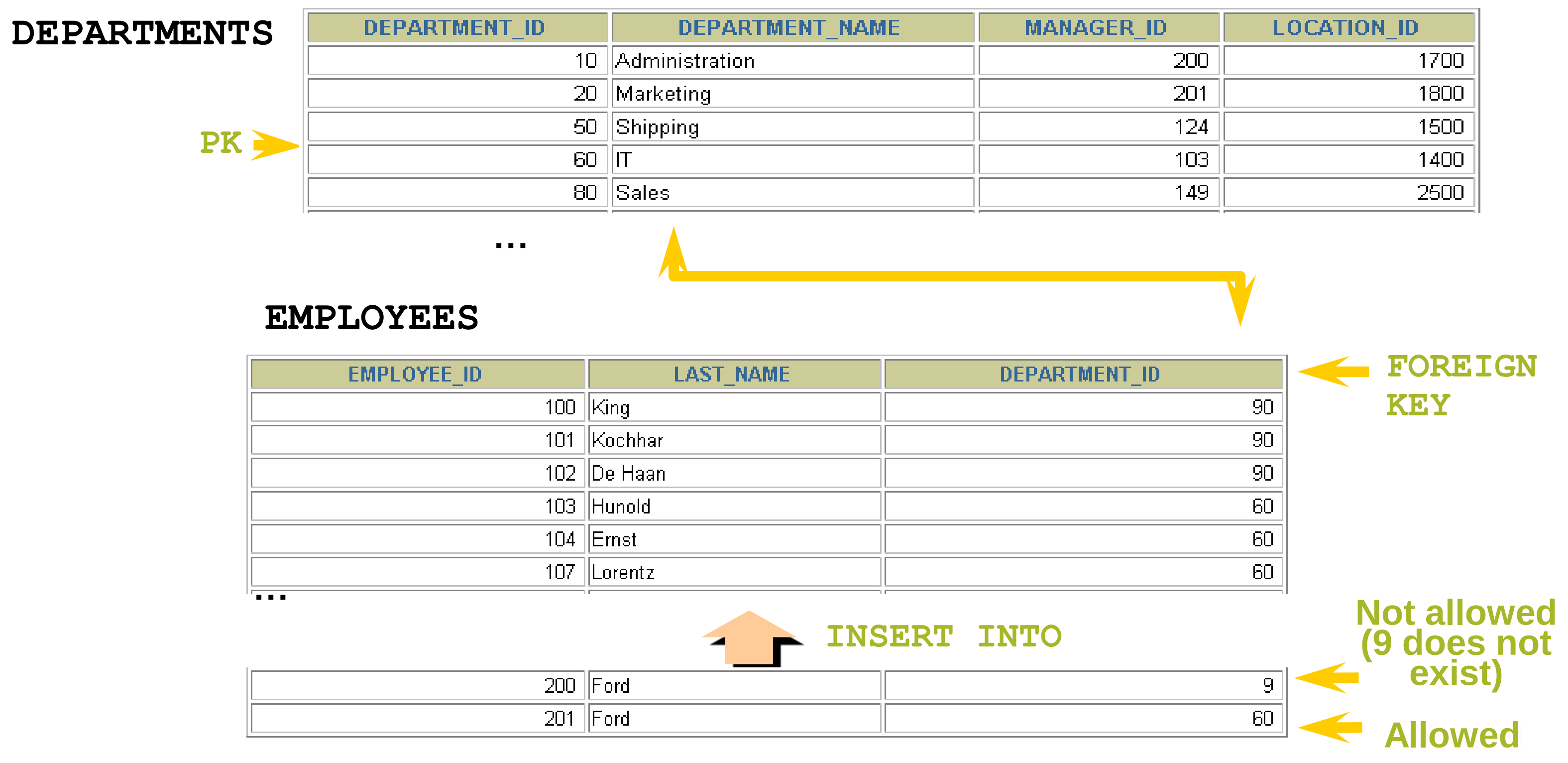
	Public Accounting		1400
50	Finance	124	1500

The PRIMARY KEY Constraint

Defined at either the table level or the column level:

```
CREATE TABLE   departments(
    department_id        NUMBER(4),
    department_name      VARCHAR2(30)
      CONSTRAINT dept_name_nn NOT NULL,
    manager_id           NUMBER(6),
    location_id          NUMBER(4),
      CONSTRAINT dept_id_pk PRIMARY KEY(department_id));
```

The FOREIGN KEY Constraint



DEPARTMENTS

DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
10	Administration	200	1700
20	Marketing	201	1800
50	Shipping	124	1500
60	IT	103	1400
80	Sales	149	2500

...

EMPLOYEES

EMPLOYEE_ID	LAST_NAME	DEPARTMENT_ID
100	King	90
101	Kochhar	90
102	De Haan	90
103	Hunold	60
104	Ernst	60
107	Lorentz	60

...

INSERT INTO

200	Ford	9
201	Ford	60

The FOREIGN KEY Constraint

Defined at either the table level or the column level:

```
CREATE TABLE employees(
    employee_id      NUMBER(6),
    last_name        VARCHAR2(25) NOT NULL,
    email            VARCHAR2(25),
    salary           NUMBER(8,2),
    commission_pct   NUMBER(2,2),
    hire_date        DATE NOT NULL,
...
    department_id    NUMBER(4),
    CONSTRAINT emp_dept_fk FOREIGN KEY (department_id)
      REFERENCES departments(department_id),
    CONSTRAINT emp_email_uk UNIQUE(email));
```

The FOREIGN KEY Constraint

- In case of defining a **foreign key** inline with column definition then you shouldn't specify FOREIGN KEY. Drop it from the definition.
- **Example:**
- CREATE TABLE **dpt**(department_id NUMBER(4), department_name VARCHAR2(30) CONSTRAINT dpt_name_nn NOT NULL, manager_id NUMBER(6), location_id NUMBER(4), CONSTRAINT dpt_id_pk PRIMARY KEY(department_id));
- CREATE TABLE **Orders**(O_Id NUMBER(4) PRIMARY KEY, OrderNo NUMBER(4) NOT NULL, department_id NUMBER(4) REFERENCES dpt(department_id));

FOREIGN KEY Constraint Keywords

- **FOREIGN KEY:** Defines the column in the child table at the table constraint level
- **REFERENCES:** Identifies the table and column in the parent table
- **ON DELETE CASCADE:** Deletes the dependent rows in the child table when a row in the parent table is deleted.
- **ON DELETE SET NULL:** Converts dependent foreign key values to null

ON DELETE CASCADE

```
CREATE TABLE supplier
( supplier_id number(10) PRIMARY KEY,
  supplier_name varchar2(50) not null,
  contact_name varchar2(50)
);


CREATE TABLE products
( product_id number(10) PRIMARY KEY,
  supplier_id number(10) not null,
  CONSTRAINT fk_supplier
    FOREIGN KEY (supplier_id)
    REFERENCES supplier(supplier_id)
    ON DELETE CASCADE
);
```

DEFAULT Constrain

- The **DEFAULT** constraint is used to insert a default value into a column.
- The default value will be added to all new records, if no other value is specified.

DEFAULT Constrain

- **DEFAULT Constraint on CREATE TABLE**
- The following SQL creates a DEFAULT constraint on the "City" column when the "Persons" table is created:

```
CREATE TABLE Persons (
    P_Id int NOT NULL,
    LastName varchar(255) NOT NULL,
    FirstName varchar(255),
    Address varchar(255),
    City varchar(255) DEFAULT 'Slemani'
)
```

The CHECK Constraint

- Defines a condition that each row must satisfy
- The following expressions are not allowed:
  - Calls to **SYSDATE**, **UID**, **USER**, and USERENV functions
  - Queries that refer to other values in other rows

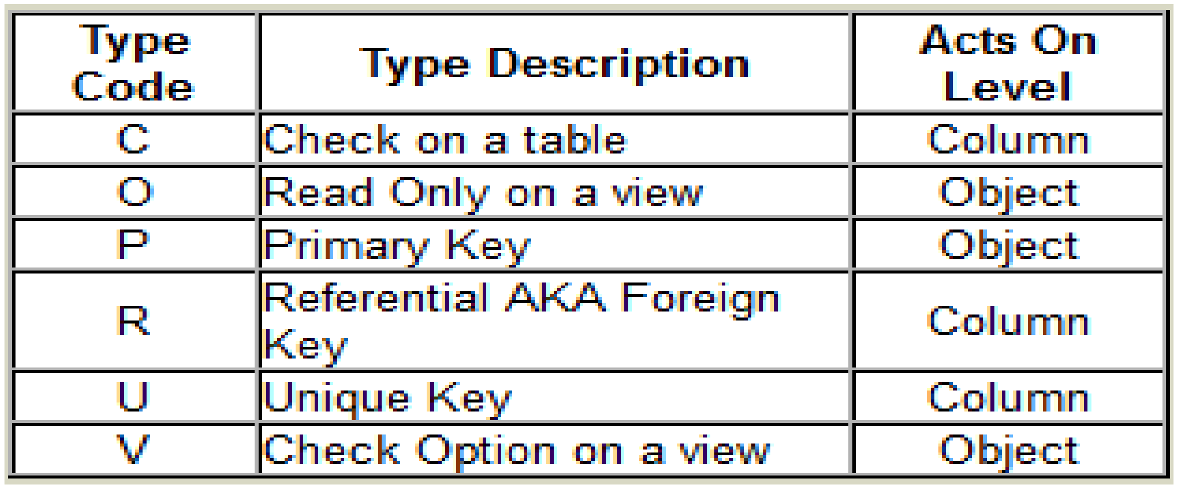
```
..., salary        NUMBER(2)
     CONSTRAINT emp_salary_min
              CHECK (salary > 0),...
```

The CHECK Constraint

-
  ```
  CREATE TABLE people (
  P_Id number(5) NOT NULL CHECK (P_Id>0),
  LastName varchar(255) NOT NULL,
  FirstName varchar(255),
   Address varchar(255),
   City varchar(255))
  ```

-
  ```
  CREATE TABLE peoples (
  P_Id number(4) NOT NULL,
  LastName varchar(255) NOT NULL,
  FirstName varchar(255),
  Address varchar(255),
  City varchar(255),
  CONSTRAINT chk_peoples CHECK (P_Id>0 AND City='Slemani'))
  ```

Constraint types in oracle



Type Code	Type Description	Acts On Level
C	Check on a table	Column
O	Read Only on a view	Object
P	Primary Key	Object
R	Referential AKA Foreign Key	Column
U	Unique Key	Column
V	Check Option on a view	Object

Viewing Constraints

Query the USER_CONSTRAINTS table to view all constraint definitions and names.

```
SELECT   constraint_name, constraint_type,
         search_condition
FROM     user_constraints
WHERE    table_name = 'EMPLOYEES';
```

CONSTRAINT_NAME	C	SEARCH_CONDITION
EMP_LAST_NAME_NN	C	"LAST_NAME" IS NOT NULL
EMP_EMAIL_NN	C	"EMAIL" IS NOT NULL
EMP_HIRE_DATE_NN	C	"HIRE_DATE" IS NOT NULL
EMP_JOB_NN	C	"JOB_ID" IS NOT NULL
EMP_SALARY_MIN	C	salary > 0
EMP_EMAIL_UK	U	

...

Viewing the Columns Associated with Constraints

View the columns associated with the constraint names in the USER_CONS_COLUMNS view.

```
SELECT   constraint_name, column_name
FROM     user_cons_columns
WHERE    table_name = 'EMPLOYEES';
```

CONSTRAINT_NAME	COLUMN_NAME
EMP_DEPT_FK	DEPARTMENT_ID
EMP_EMAIL_NN	EMAIL
EMP_EMAIL_UK	EMAIL
EMP_EMP_ID_PK	EMPLOYEE_ID
EMP_HIRE_DATE_NN	HIRE_DATE
EMP_JOB_FK	JOB_ID
EMP_JOB_NN	JOB_ID

...

Enabling and Disabling Constraints

```
ALTER TABLE table_name
DISABLE CONSTRAINT constraint_name;
```

```
ALTER TABLE table_name
ENABLE CONSTRAINT constraint_name;
```

Modifying table for constraint

- Constraints can not be altered. They must be dropped and recreated.
- Some modifications are allowed via ALTER TABLE.
- **Drop Syntax** :

  Alter table TableName drop constraint ConstraintName
- **Add Syntax**:

  Alter table TableName add constraint ConstraintName
- **Example**:

  Alter table test add constraint ck_cn check(price > 5);
- **In case of existing data inside the table write:**

  Alter table test add constraint ck_cn check(price > 5) enable novalidate;

Example

- **Adding check constraint:**

```
ALTER TABLE supplier
ADD CONSTRAINT check_supplier_name CHECK (supplier_name IN ('IBM',
'Microsoft', 'NVIDIA'));
```

- **Adding primary key:**

```
ALTER TABLE supplier
ADD CONSTRAINT supplier_pk PRIMARY KEY (supplier_id);
```

Any Question?