CSE-302
Database Management Systems
Sessional

CONSTRAINTS

Constraints

- Constraints are rules to enforce business rules, practices, and policies

- **Why do we need constraints?**
  - To keep the database reliable.
  - To prevent a user from entering non-sensical data.
  - The business or other organization has certain rules that cannot be violated.

- Constraints are used for implementing the rules.

Reasons for using Constraints

- Enforce rules at the table level whenever a row is **inserted, updated** or **deleted** from the table. The constraints must be satisfied for the operation to be succeed.

- **Prevent** the **deletion** of a table if there are **dependencies** from other tables.

- Provide **rules** from Oracle tools such as Oracle Developer.

Types of Constraints

Constraint	Abbr.	Description
UNIQUE	_uk	•Ensures that all data values stored in a specific column are unique. •It differs from the PK in that it allows NULL values.
NOT NULL	_nn	•Ensures that a specified column can not contain any NULL value. •It can only be created in the column level approach to table creation.
PRIMARY KEY	_pk	•Determine which column(s) uniquely identifies each record. •It can not be NULL. •Data values must be unique.
CHECK	_ck	•Ensures that a specified condition is true before the data value is added to the table.
FOREIGN KEY	_fk	•In a one-to-many relationship, it is added to the 'many' table. •The constraint ensures that if a value is inserted into a specified column, it must already exist in the 'one' table, or the record is not added.

Ways of applying Constraints

- As part of a **CREATE TABLE** command

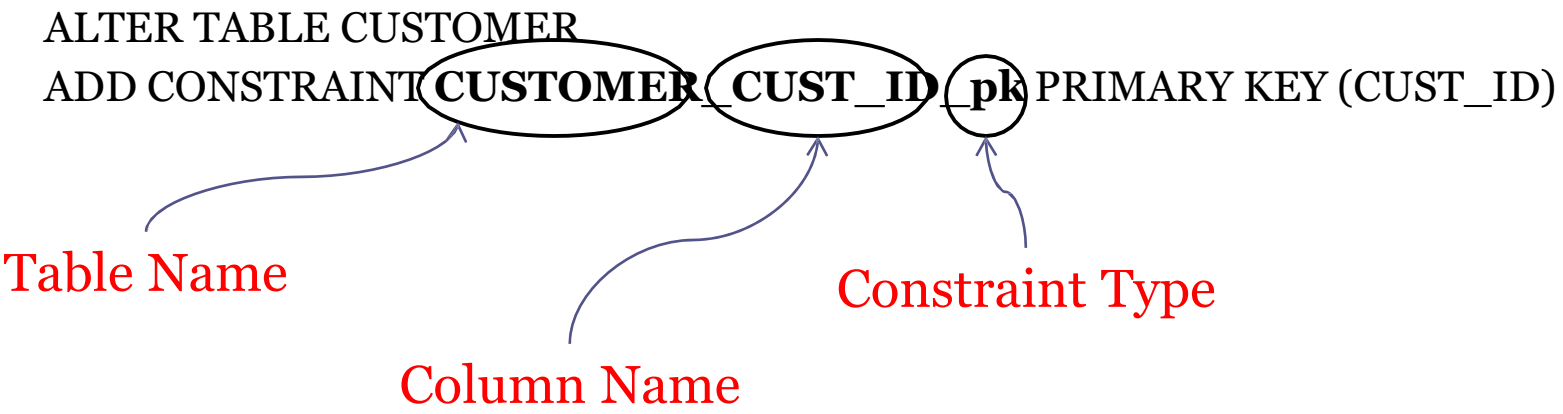
or

- As part of an **ALTER TABLE** command

Syntax for entering a constraint name

TableName_ColumName_ConstraintType

Apply the Primary Key constraint on the CUST_ID column of Customer table.



- Any constraint can be created at column level or at the table level

PRIMARY KEY (Table Level)

```
Create table Customer
(
    Cust_id VARCHAR2(12) ,
    Cust_name VARCHAR2(12),
    Cust_dob DATE,
    Cust_street  VARCHAR2(12),
    Cust_city  VARCHAR2(12),
    CONSTRAINT Customer_CUST_ID_pk PRIMARY KEY(CUST_ID)
);
```

```
ALTER TABLE CUSTOMER
ADD CONSTRAINT Customer_CUST_ID_pk PRIMARY KEY(CUST_ID);
```

```
ALTER TABLE CUSTOMER
ADD PRIMARY KEY(CUST_ID);
```

PRIMARY KEY (Column Level)

```
Create table Customer
(
    Cust_id VARCHAR2(12) CONSTRAINT Customer_CUST_ID_pk PRIMARY KEY,
    Cust_name VARCHAR2(12),
    Cust_dob DATE,
    Cust_street  VARCHAR2(12),
    Cust_city  VARCHAR2(12)
);
```

```
Create table Customer
(
    Cust_id VARCHAR2(12) PRIMARY KEY,
    Cust_name VARCHAR2(12),
    Cust_dob DATE,
    Cust_street  VARCHAR2(12),
    Cust_city  VARCHAR2(12)
);
```

PRIMARY KEY -COMPOSITE

- Simply list the column names within parentheses after the constraint type.

```
ALTER TABLE orderitems
ADD CONSTRAINT orderitems_pk PRIMARY KEY (order_id, item_no);
```

Table Name

Constraint Type

Multiple Column Names

- After this constraint is added to the ORDERITEMS table, a user can enter only a unique combination of Order_id and Item_no for each new row.

UNIQUE (Table Level)

```
Create table Account
 (
        Account_id VARCHAR2(12),
        Balance NUMBER(20,5),
        Type VARCHAR2(8),
        CONSTRAINT Account_ACCOUNT_ID_uk UNIQUE (ACCOUNT_ID)
);
```

```
ALTER TABLE ACCOUNT
ADD CONSTRAINT Account_ACCOUNT_ID_uk UNIQUE (ACCOUNT_ID);
```

```
ALTER TABLE CUSTOMER
ADD UNIQUE (ACCOUNT_ID);
```

UNIQUE (Column Level)

```
Create table Account
(
    Account_id VARCHAR2(12) CONSTRAINT Account_ACCOUNT_ID_uk UNIQUE,
    Balance NUMBER(20,5),
    Type VARCHAR2(8)

);
```

```
Create table Account
(
    Account_id VARCHAR2(12) UNIQUE ,
    Balance NUMBER(20,5),
    Type VARCHAR2(8)

);
```

UNIQUE vs Primary Key

- A **UNIQUE** constraint allows **NULL** values unless define **NOT NULL** in the same column
- A **PRIMARY KEY** constraint does not allow **NULL** values

CHECK (Table Level)

```
Create table Account
 (
        Account_id VARCHAR2(12),
        Balance NUMBER(20,5),
        Type VARCHAR2(8),
        CONSTRAINT Account_BALANCE_ck CHECK (Balance>0)
);
```

```
ALTER TABLE ACCOUNT
ADD CONSTRAINT Account_BALANCE_ck CHECK (Balance>0);
```

```
ALTER TABLE CUSTOMER
ADD CHECK (Balance>0);
```

CHECK (Column Level)

```
Create table Account
(
    Account_id VARCHAR2(12),
    Balance NUMBER(20,5) CONSTRAINT Account_BALANCE_ck CHECK (Balance>0),

    Type VARCHAR2(8)

);
```

```
Create table Account
(
    Account_id VARCHAR2(12) ,
    Balance NUMBER(20,5) CHECK (Balance>0),
    Type VARCHAR2(8)

);
```

NOT NULL(Table Level)

```
ALTER TABLE CUSTOMER
MODIFY CUST_ID CONSTRAINT Customer_CUST_ID_nn NOT NULL;
```

```
ALTER TABLE CUSTOMER
MODIFY CUST_ID NOT NULL;
```

NOT NULL (Column Level)

```
Create table Customer
(
    Cust_id VARCHAR2(12) CONSTRAINT Customer_CUST_ID_nn NOT NULL,
    Cust_name VARCHAR2(12),
    Cust_dob DATE,
    Cust_street  VARCHAR2(12),
    Cust_city  VARCHAR2(12)
);
```

```
Create table Customer
(
    Cust_id VARCHAR2(12) NOT NULL,
    Cust_name VARCHAR2(12),
    Cust_dob DATE,
    Cust_street  VARCHAR2(12),
    Cust_city  VARCHAR2(12)
);
```

FOREIGN KEY

- The FOREIGN KEY constraint is used to prevent actions that would destroy links between tables.
- A FOREIGN KEY is a field (or collection of fields) in one table, that refers to the PRIMARY KEY in another table.
- The table with the foreign key is called the **child table**, and the table with the primary key is called the **referenced or parent table**.

Student (Parent Table)

PK

Roll_no	Name	Address
1	A	Dhaka
2	B	Sylhet

PK **Course** (Child Table) FK

Course_id	Course_Name	Roll_no
CSE 302	Database	1

FOREIGN KEY (Table Level)

```
Create table Course
(
    Course_id VARCHAR2(12) ,
    Course_name  VARCHAR2(12),
    Roll_no  INT,
    CONSTRAINT Course_COURSE_ID_pk  PRIMARY KEY (COURSE_ID),
    CONSTRAINT  Course_Roll_NO_fk  FOREIGN KEY (Roll_no)
    REFERENCES Student (Roll_no)
    );
```

```
ALTER TABLE CUSTOMER
ADD CONSTRAINT  Course_Roll_NO_fk  FOREIGN KEY (Roll_no)
REFERENCES Student (Roll_no);
```

```
ALTER TABLE CUSTOMER
ADD FOREIGN KEY (Roll_no) REFERENCES Student (Roll_no);
```

FOREIGN KEY (Column Level)

```
Create table Course
(
    Course_id VARCHAR2(12) CONSTRAINT Course_COURSE_ID_pk PRIMARY KEY,
    Course_name VARCHAR2(12),
    Roll_no  INT CONSTRAINT  Course_Roll_NO_fk  FOREIGN KEY  REFERENCES Student
    (Roll_no)
);
```

```
Create table Course
(
    Course_id VARCHAR2(12) PRIMARY KEY,
    Course_name VARCHAR2(12),
    Roll_no  INT FOREIGN KEY REFERENCES Student (Roll_no)
);
```

FOREIGN KEY

PK **Student** (Parent Table)

Roll_no	Name	Address
1	A	Dhaka
2	B	Sylhet

Course (Child Table) FK

Course_id	Course_Name	Roll_no
CSE 302	Database	1

- **Insert** – No violation
- **Delete** – May cause violation

Solution: Use **ON DELETE CASCADE**

- **Update** – May cause violation

Solution: Use **ON UPDATE CASCADE**

- **Insert** – May cause violation
- **Delete** – No violation
- **Update** – May cause violation

FOREIGN KEY
(ON DELETE CASCADE/ ON UPDATE CASCADE)

```
Create table Course
(
    Course_id VARCHAR2(12) ,
    Course_name  VARCHAR2(12),
    Roll_no VARCHAR2(12),
    CONSTRAINT Course_COURSE_ID_pk  PRIMARY KEY (COURSE_ID),
    CONSTRAINT  Course_Roll_NO_fk  FOREIGN KEY (Roll_no)
    REFERENCES Student (Roll_no) ON DELETE CASCADE
    );
```

```
Create table Course
(
    Course_id VARCHAR2(12) ,
    Course_name  VARCHAR2(12),
    Roll_no  INT,
    CONSTRAINT Course_COURSE_ID_pk  PRIMARY KEY (COURSE_ID),
    CONSTRAINT  Course_Roll_NO_fk  FOREIGN KEY (Roll_no)
    REFERENCES Student (Roll_no) ON UPDATE CASCADE
    );
```

FOREIGN KEY - Composite

```
CREATE TABLE Depositor
(
   Cust_id VARCHAR2(12) NOT NULL,
   Account_id VARCHAR2(12) NOT NULL,

   COSNTRAINT DEPOSITOR_CUST_ID_FK FOREIGN
   KEY(CUST_ID) REFERENCES CUSTOMER(CUST_ID),

    COSNTRAINT DEPOSITOR_ACCOUNT_ID_FK FOREIGN
   KEY(ACCOUNT_ID) REFERENCES ACCOUNT
   (ACCOUNT_ID)
);
```

DROP Constraints

- To drop a constraint, you can identify the constraint name from the USER_CONSTRAINTS and then use ALTER TABLE command with the DROP clause.

ALTER TABLE **Table_Name**
DROP CONSTRAINT **Constraint_Name**;

- To remove the primary key constraint from the Customer Table and drop the associated FOREIGN KEY constraint-

ALTER TABLE **Table_Name**
DROP CONSTRAINT **Constraint_Name**
CASCADE;

Viewing constraints

- Query the USER_CONSTRAINTS table to view all the constraint definition and names.

```
SELECT CONSTRAINT_NAME, CONSTRAINT_TYPE
   FROM USER_CONSTRAINTS
WHERE TABLE_NAME='CUSTOMER';
```

- Viewing The Columns Associated With Constraints

```
SELECT
CONSTRAINT_NAME,COLUMN_NAME  FROM
USER_CONS_COLUMNS
WHERE TABLE_NAME='CUSTOMER';
```

Practice Problems for Constraints

- **CREATE TABLE BORROWER in such a way that Cust_ID must be in Customer table and Loan_ID must be in LOAN table.**

THANK YOU