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
PRIMARY KEY	_pk	•Determine which column(s) uniquely identifies each record. •It can not be NULL. •Data values must be unique.
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.
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.
CHECK	_ck	•Ensures that a specified condition is true before the data value is added to the table.
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.

Ways of applying Constraints

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

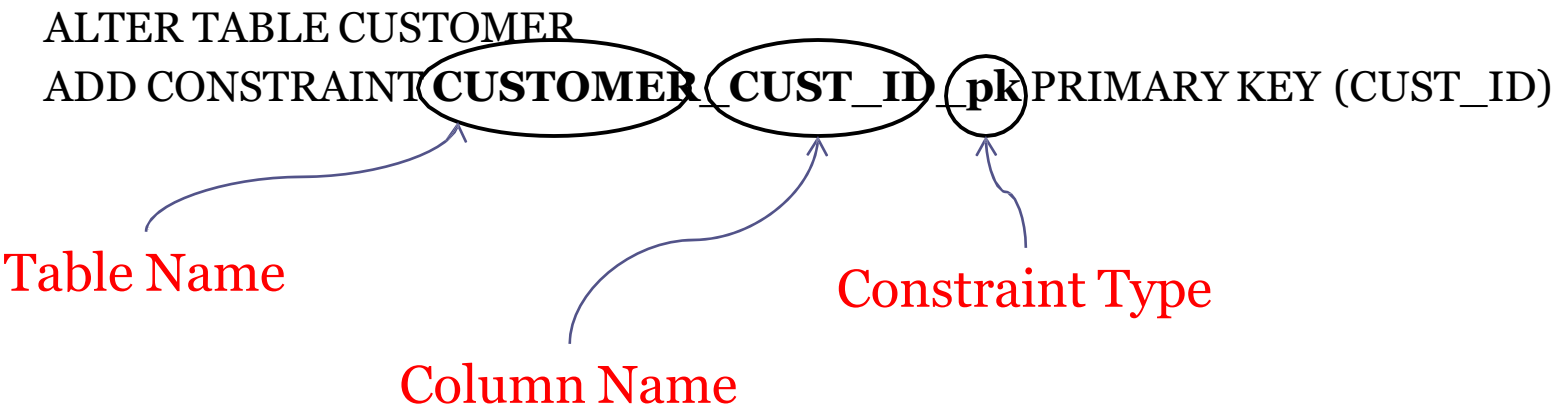
or

- 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#, item#);
```

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# and Item# for each new row.

NOT NULL

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

NOT NULL

ALTER TABLE CUSTOMER
MODIFY (CUST_ID **NOT NULL**);

or

ALTER TABLE CUSTOMER
MODIFY
(CUST_ID **CONSTRAINT Customer_CUST_ID_nn NOT NULL**);

UNIQUE

```
Create table Account  (
Account_id VARCHAR2(12) NOT NULL ,
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);
```

UNIQUE

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

FOREIGN KEY

```
ALTER TABLE Depositor
ADD CONSTRAINT Depositor_Cust_ID_fk FOREIGN KEY (Cust_ID)
    REFERENCES Customer (Cust_ID);
```

- A record cannot be deleted in the parent table (CUSTOMER) if matching entries exist in the child table.
- That is, you cannot delete a customer form the CUSTOMERS table if there are Account in the DEPOSITOR table that Customer.
- But what if you really want to remove a customer (from the CUSTOMERS table) that does have related Account (in the DEPOSITOR table).

FOREIGN KEY

- **The conventional method is to**
  - First, remove the related records from the child table (DEPOSITOR)
  - Then, remove the customer record form the CUSTOMER table.

- **A simpler method is available:**

```
ALTER TABLE DEPOSITOR
ADD CONSTRAINT DEPOSITOR _CUST_ID_fk
FOREIGN KEY (CUST_ID) REFERENCES CUSTOMER (CUST_ID) ON
    DELETE CASCADE;
```

- **If a record is deleted from the parent table, then any corresponding records in the child table are also automatically deleted.**
  - To try the above, you have to first remove the original FOREIGN KEY constraint:

```
ALTER TABLE DEPOSITOR
DROP CONSTRAINT DEPOSITOR _CUST_ID_fk;
```

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)
);
```

CHECK

```
Create table Account
(
Account_id VARCHAR2(12) NOT NULL UNIQUE,
Balance NUMBER(20,5) CHECK( Balance>0),
Type VARCHAR2(8)
);

Create table Account
(
Account_id VARCHAR2(12) NOT NULL UNIQUE,
Balance NUMBER(20,5),
Type VARCHAR2(8),
CONSTRAINT Account_Balance_ck  CHECK(Balance>0)
);
```

ADD Constraints

- You can add, drop, enable or disable a constraint, but you cannot modify its structure.

- You can add a NOT NULL constraint to an existing column by using the MODIFY Clause of the ALTER TABLE statement.

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 DEPOSITOR
DROP CONSTRAINT
DEPOSITOR_CUST_ID_fk;
```

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

```
ALTER TABLE CUSTOMER
DROP PRIMARY KEY CASCADE;
```

Viewing constraints

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

```
SELECT CONSTRAINT_NAME, CONSTRAINT_TYPE,
    SEARCH_CONDITION
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