DATA CONSTRAINT

Prepared By: Dr. Vipul Vekariya
What is constraint?

- Constraints enforce rules at the table level.
- Constraints prevent the deletion of a table if there are dependencies.

- There are two types of data constraint.
  1) I/O constraint
  2) business rule constraint
The Primary Key Constraint

- A primary key is used to uniquely identify each row in a table.
- It may be one or more columns in a table.
- A table can have only one primary key.
- A primary key column in a table has special attributes.
- It defines the column as a mandatory column (column cannot be left blank). As the not null attribute active.
- The data held across the column must be unique.
- A single column primary key is called simple key.
- A multicolunm primary key is called a composite primary key.
Features of primary key

- It used for uniquely identify a row.
- It will not allow the duplicate values.
- It will not allow the null values.
- It is not compulsory but it is recommended.
- Only one primary key allowed per table.
- One table can combine up to 16 columns in a composite primary key.
Syntax of primary key

- `<column name> <data type>(<size>) primary key`
- Primary key at column level.
- Example:

```sql
CREATE TABLE CUST_MSTR (
"CUST_NO" VARCHAR2(10) PRIMARY KEY,
"FNAME" VARCHAR2(25),
"MNAME" VARCHAR2(25),
"FORM60" VARCHAR2(1));
```
Primary key at table level

Example:

```sql
CREATE TABLE FD_MSTR(
"FD_SER_NO" VARCHAR2(10),
"MANAGER_SIGN" VARCHAR2(1),
PRIMARY KEY(FD_SER_NO, CORP_CUST_NO));
```
The Foreign key

- Foreign key represent relationship between two tables.
- It is a column or group of column whose value are derived from the primary key or unique key of some other table.
- The table in which foreign key is defined is called foreign table.
- The table that defines the primary or unique key and is referenced by foreign key is called the primary table.
- It defines at create table statement or alter table statement.
Features of Foreign Key

- Foreign key is column that references a column of a table.
- Parent record can be delete provided if no child record exit.
- Master table cannot be updated if child record exit.
- This constraint establish relationship between records(column data). Across master and detail table.
- Record can not be inserted into detail table if corresponding record in master table do not exist.
- Record of the master table cannot be deleted if the corresponding record in the detail table actually exist.
Syntax of foreign key

- `<column name>` `<data type>`(<`size`>)
  REFERENCES `<table name>` [(`<column name>`)]

Example:

```sql
CREATE TABLE EMP_MSTR(
    "EMP_NO" VARCHAR2(10) PRIMARY KEY,
    "BRANCH_NO" VARCHAR2(10) REFERENCES BRANCH_MSTR,
    "FNAME" VARCHAR2(25),
    "DESIG" VARCHAR2(30));
```
Foreign key at table level

- foreign key (<column name> [,<column name>] )
  references <table name> [(<column name> ,<column name>))]

Example:

CREATE TABLE ACCT_FD_CUST_DTLS(
    "ACCT_FD_NO" VARCHAR2(10),
    "CUST_NO" VARCHAR2(10),
    FOREIGN KEY (CUST_NO) REFERENCES CUST_MSTR(CUST_NO));
Assigning user defined name to constraints

- CREATE TABLE departments
  (department_id NUMBER(4),
   department_name VARCHAR2(30),
   manager_id NUMBER(6),
   location_id NUMBER(4),
   CONSTRAINT dept_id_pk PRIMARY KEY(department_id));
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));
Adding a Constraint Syntax

- Use the ALTER TABLE statement.
- Add or drop a constraint, but not modify its structure
- Enable or disable constraints

```
ALTER TABLE tablename
    ADD [CONSTRAINT NAME] constraint type (column);
```

```
ALTER TABLE employees
    ADD PRIMARY KEY(employee_no);
```

Table altered.
Dropping a Constraint

- Remove the manager constraint from the EMPLOYEES table.
- ALTER TABLE employees
  DROP CONSTRAINT emp_manager_fk;
- Table altered
- ALTER TABLE departments
  DROP PRIMARY KEY;
- Table altered.
Disabling Constraints

- Execute the DISABLE clause of the ALTER TABLE statement to deactivate an integrity constraint.
- Example:
  ```sql
  ALTER TABLE employees
  DISABLE CONSTRAINT emp_id_pk;
  ```
- Table altered.
Enabling Constraints

- Activate an integrity constraint currently disabled in the table definition by using the ENABLE clause.

- Example:

  ```sql
  ALTER TABLE employees
  ENABLE CONSTRAINT emp_id_pk;
  ```

- Table altered.
Viewing Constraints

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

- Example:

  ```sql
  SELECT constraint_name, constraint_type, search_condition
  FROM user_constraints
  WHERE table_name = 'branch_mstr';
  ```
Result of Previous query

<table>
<thead>
<tr>
<th>CONSTRAINT_NAME</th>
<th>C</th>
<th>SEARCH_CONDITION</th>
</tr>
</thead>
<tbody>
<tr>
<td>EMP_LAST_NAME_NN</td>
<td>C</td>
<td>&quot;LAST_NAME&quot; IS NOT NULL</td>
</tr>
<tr>
<td>EMP_EMAIL_NN</td>
<td>C</td>
<td>&quot;EMAIL&quot; IS NOT NULL</td>
</tr>
<tr>
<td>EMP_HIRE_DATE_NN</td>
<td>C</td>
<td>&quot;HIRE_DATE&quot; IS NOT NULL</td>
</tr>
<tr>
<td>EMP_JOB_NN</td>
<td>C</td>
<td>&quot;JOB_ID&quot; IS NOT NULL</td>
</tr>
<tr>
<td>EMP_SALARY_MIN</td>
<td>C</td>
<td>salary &gt; 0</td>
</tr>
<tr>
<td>EMP_EMAIL_UK</td>
<td>U</td>
<td></td>
</tr>
</tbody>
</table>
Viewing the Columns Associated with Constraints

- View the columns associated with the constraint names in the USER_CONS_COLUMNS view.
- Example:

  ```sql
  SELECT constraint_name, column_name
  FROM user_cons_columns
  WHERE table_name = 'EMPLOYEES';
  ```
Result of Previous query

<table>
<thead>
<tr>
<th>CONSTRAINT_NAME</th>
<th>COLUMN_NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>EMP_DEPT_FK</td>
<td>DEPARTMENT_ID</td>
</tr>
<tr>
<td>EMP_EMAIL_NN</td>
<td>EMAIL</td>
</tr>
<tr>
<td>EMP_EMAIL_UK</td>
<td>EMAIL</td>
</tr>
<tr>
<td>EMP_EMP_ID_PK</td>
<td>EMPLOYEE_ID</td>
</tr>
<tr>
<td>EMP_HIRE_DATE_NN</td>
<td>HIRE_DATE</td>
</tr>
<tr>
<td>EMP_JOB_FK</td>
<td>JOB_ID</td>
</tr>
<tr>
<td>EMP_JOB_NN</td>
<td>JOB_ID</td>
</tr>
</tbody>
</table>
Unique key constraint

- Unique key constraint permits the multiple NULL into column.
- The NULL values are clubbed at the top of the column.
- This is difference between primary key and unique key.
- Unique key will not allow duplicate values.
- Unique index is created automatically.
- A table can have more than one unique key which is not possible in primary key.
- Unique key can combine up to 16 columns in a composite unique key.
- Unique key can not be LONG or LONG RAW data type.
Example of unique key.

- CREATE TABLE CUST_MSTR (  
  "CUST_NO" VARCHAR2(10) UNIQUE,  
  "FNAME" VARCHAR2(25),  
  "MNAME" VARCHAR2(25),  
  "LNAME" VARCHAR2(25),  
  "DOB_INC" DATE,  
  "OCCUP" VARCHAR2(25),  
  "PHOTOGRAPH" VARCHAR2(25),  
  "SIGNATURE" VARCHAR2(25),  
  "PANNO" VARCHAR2(1) UNIQUE,  
  "FORM60" VARCHAR2(1));
Business Rule Constraints

- Check Constraint:

  Syntax: CHECK (<logical expression>)

Example:

```sql
CREATE TABLE CUST_MSTR(
    "CUST_NO" VARCHAR2(10) CHECK (CUST_NO LIKE 'C%'),
    "FNAME" VARCHAR2(25) CHECK (FNAME = upper(Fname)),
    "MNAME" VARCHAR2(25) CHECK (MNAME = upper(Mname)),
    "LNAME" VARCHAR2(25) CHECK (LNAME = upper(Lname)),
    "DOB_INC" DATE,
    "OCCUP" VARCHAR2(25),
    "PHOTOGRAPH" VARCHAR2(25),
    "SIGNATURE" VARCHAR2(25),
    "PANCOPY" VARCHAR2(1));
```
NULL values Concept

- Setting NULL value is appropriate when the actual value is unknown.
- A NULL value is not equivalent to zero or space.
- A NULL value will evaluate to NULL if any expression.
- NULL value can be inserted into column of any data type.
- If the column has NULL value, oracle ignores any constraint that match with that column.
NOT NULL constraints

- NOT NULL concept means column can not be left empty.
  
syntax:  `<column name> < data type>(<size>) NOT NULL`

Example:

- 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 NOT NULL);
Default value

- Syntax:
  `<column name> < data type> (<size>)  Default <value>;`

Example:

```sql
CREATE TABLE ACCT_MSTR 
("ACCT_NO" VARCHAR2(10),
 "SF_NO" VARCHAR2(10),
 "LF_NO" VARCHAR2(10),
 "BRANCH_NO" VARCHAR2(10),
 "INTRO_CUST_NO" VARCHAR2(10),
 "STATUS" VARCHAR2(1) DEFAULT 'A');
```