Constraints

SCS2209-Database II
Mr. Rangana Jayashanka

Data Integrity

- Ensures that the values entering to the database is accurate and valid.
- Uses integrity constraints.
- Integrity constraints maintains accurate databases by eliminating invalid data updates/ insert/ deletes.

eid	Ename	Designation	Salary	did	dname	location
1000	Chamath	Lecturer	60000	1	Academic	CMB
1001	Viraj	Executive	45000	3	Maintenance	SJP
1002	Manju	Lecturer	75000	1	Academic	CMB
~~1003~~	Kasun	Manager	50000	2	Admin	RHN
1004	Ishani	Lecturer	35000	1	Academic	CMB
1005	Randil	Lecturer	80000	1	Academic	CMB
~~1006~~	Thilini	Assistant	25000	2	Admin	RHN
1007	Roshan	Lecturer	42000	1	Academic	CMB
1008	Supun	Assistant	28000	4	NOC	CMB

1009	**Upul**	**Lecturer**	**35000**	**16**	**Academic**	**CMB**

Database Integrity Constraints

- ➢ Constraints are conditions that specify restrictions on the database state.
- ➢ Types in relational DB

- **Entity integrity** does not allow two rows with the same identity in a table.
- **Domain integrity** allows only predefined values.
- **Referential integrity** allows only the consistency of values across related tables.
- **User-defined integrity** define constraints.

Database Constraints

- ➢ They are the restrictions on the contents of the database and its operations.
- ➢ Types of Constraints:

  - Primary key constraint
  - Foreign key constraint (referential integrity)
  - Unique constraint
  - Not Null constraint
  - Check constraint
  - Default constraint

Primary Key Constraints

- ***Primary key*** uniquely identifies each record in a table.
- It must have unique values and cannot contain nulls.
  - This is because primary key values are used to identify the individual tuples.
  - If PK has several attributes, null is not allowed in any of these attributes.

- Table can have only one primary key.
- In the below example the **studentId** field is marked as primary key, that means the **studentId** field cannot have duplicate and null values.

Primary Key Constraints

```
CREATE TABLE  Student(
studentId     CHAR(10),
name          CHAR(20),
address       CHAR(25),
age           INT,
CONSTRAINT pk_stdID PRIMARY KEY (studentId));
```

Unique Constraint

- UNIQUE constraint enforces a column or set of columns to have unique values.
- If a column has a unique constraint, it means that particular column cannot have duplicate values in a table.
- A PRIMARY KEY constraint automatically has a UNIQUE constraint.

Unique Constraint

```
CREATE TABLE Student(
Stdid       CHAR(10)PRIMARY KEY,
Name        CHAR(20),
Address     CHAR(25),
Age         INT,
NIC         CHAR (10) UNIQUE
)
```

Not Null Constraint

- NOT NULL constraint makes sure that a column does not hold NULL value.
- When we don’t provide value for a particular column while inserting a record into a table, it takes NULL value by default.
- By specifying NULL constraint, we can be sure that a particular column(s) cannot have NULL values.

Not Null Constraint

```
CREATE TABLE Student (
Sid INT Primary Key,
name    CHAR(20) NOT NULL,
address CHAR(25),
age INT
);
```

DEFAULT Constraint

- The DEFAULT constraint provides a default value to a column when there is no value provided while inserting a record into a table.

DEFAULT Constraint

```
CREATE TABLE Student (
name        CHAR (20),
address     CHAR (25),
Department CHAR (20) DEFAULT "Computer Science",
Age         INT
);
```

Check Constraint

- This constraint is used for specifying range of values for a particular column of a table.
- When this constraint is being set on a column, it ensures that the specified column must have the value falling in the specified range.

Check Constraint

```
CREATE TABLE UnderGrad_student (
sid             CHAR (25) Primary Key,
name            CHAR (20),
address         CHAR (25),
Age             INT,
Reg_Course      CHAR(10) CHECK (Age BETWEEN 19 and 26)
);
```

Domain Constraint

- Each table has certain set of columns, and each column allows a same type of data, based on its data type.
- The column does not accept values of any other data type.
- Domain constraints are user defined data type and we can define them like this:
- Domain Constraint = data type + Constraints (NOT NLL / UNIQUE / PRIMATY KEY/ FOREUGN KEY / CHECK/ DEFAULT)

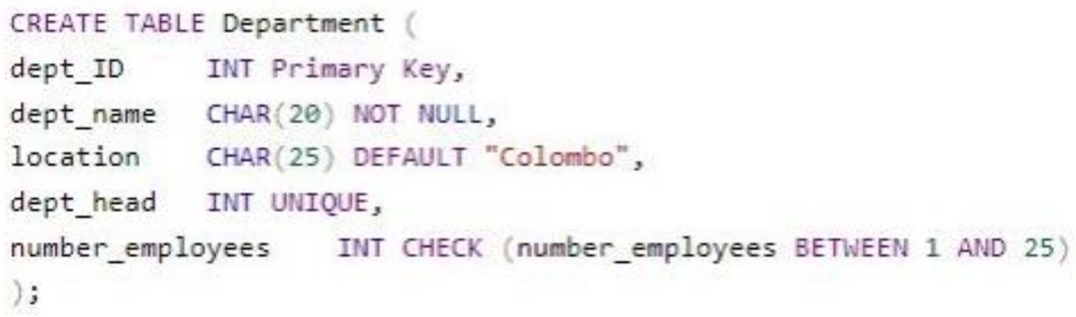
Activity

Create a table called “Department” with the following constraints.

- ➢ Dept_ID is a number used as the primary key.
- ➢ dept_name cannot be null
- ➢ I want default location to be ‘Colombo’
- ➢ dept_head specifies a unique number
- ➢ number_employees should be an integer between 1-25

Activity-Answer

- ➢ dept_ID is a number used as the primary key
- ➢ dept_name cannot be null
- ➢ I want default location to be 'Colombo'
- ➢ dept_head specifies a unique number
- ➢ number_employees should be an integer between 1-25



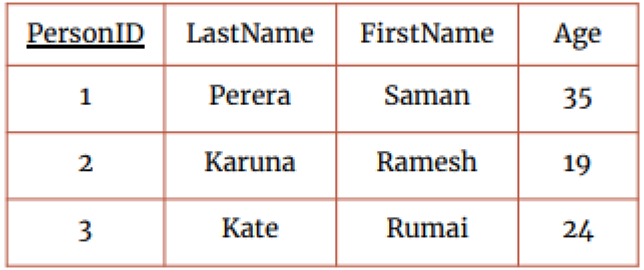
```
CREATE TABLE Department (
dept_ID       INT Primary Key,
dept_name     CHAR(20) NOT NULL,
location      CHAR(25) DEFAULT "Colombo",
dept_head     INT UNIQUE,
number_employees     INT CHECK (number_employees BETWEEN 1 AND 25)
);
```

Foreign Key Constraint (referential integrity)

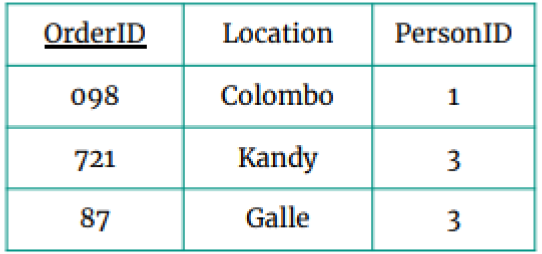
- A FOREIGN KEY is a key used to link two tables together.
- Foreign keys are the columns of a table that points to the primary key (unique) of another table.
- They act as a cross-reference between tables.
- The table containing the foreign key is called the *child table/ referencing table,* and the table containing the candidate key is called the *referenced or parent table.*

Foreign Key Constraint (referential integrity)

- The FOREIGN KEY constraint prevents invalid data from being inserted into the foreign key column.
- It has to be one of the values contained in the table it points to.



PersonID	LastName	FirstName	Age
1	Perera	Saman	35
2	Karuna	Ramesh	19
3	Kate	Rumai	24



OrderID	Location	PersonID
098	Colombo	1
721	Kandy	3
87	Galle	3

Foreign Key Constraint

```
CREATE TABLE Orders (
OrderID INT NOT NULL,
location        CHAR (25),
personID        INT,
PRIMARY KEY (OrderID),
CONSTRAINT FK_PersonOrder FOREIGN KEY (PersonID),
REFERENCES Persons (PersonID)
);
```

Referential Triggered Action

- ➢ Updates may propagate to cause other updates automatically.
- ➢ Operations
  - ON DELETE
  - ON UPDATE

- ➢ Actions To Take
  - RESTRICT: Reject the row to be deleted.
  - SET NULL: Set value of foreign key to NULL.
  - SET DEFAULT: Set value of foreign key to default value.
  - CASCADE: Delete/ Update referencing row(s) as well.
  - NO ACTION

Violations when INSERT/UPDATE

- Domain constraint Violation: If one of the attribute values provided for the new tuple is not of the specified attribute domain.
- Key constraint Violation: if the value of a key attribute in the new tuple already exists in another tuple in the relation.
- Entity integrity Violation: if the primary key value is null in the new tuple.
- Referential integrity Violation: If a foreign key value in the new tuple references a primary key value that does not exist in the referenced relation.

Example

Fname	Lname	<u>Ssn</u>	Bdate	Address	Sex	Salary	Super_Ssn	Dno
Kasun	Perera	234532	1999-12-13	Colombo	M	230000	343534	7
Shiva	Krishan	89892	2000-08-23	Kandy	M	78200	32149	5
Ameena	Safran	43422	2008-07-12	Gampaha	F	82300	89943	1
Stephani	Shaw	21898	2000-09-28	Galle	F	23000	78687	2

1. Insert <’Sama’, ‘Jayasena’,Null,’1989-05-27’,’Matara’,’F’,67000, Null,4> into EMPLOYEE

   Entity integrity Violation

2. Insert <’Mary’, ‘Doe’,234532,’2004-05-27’,’Badulla’,’F’,98000, NULL,4> into EMPLOYEE

   Key constraint Violation

3. Insert <’Raj’,’Kumaran’,’345454’,’1989’,NULL,’M’,’100k’, Null,4> into EMPLOYEE

   Domain constraint Violation

Example

Fname	Lname	Ssn	Bdate	Address	Sex	Salary	Super_Ssn	Dno
Kasun	Perera	234532	1999-12-13	Colombo	M	230000	343534	7
Shiva	Krishan	898921	2000-08-23	Kandy	M	78200	32149	5
Ameena	Safran	434221	2008-07-12	Gampaha	F	82300	89943	1
Stephani	Shaw	218983	2000-09-28	Galle	F	23000	78687	2

```
4.  Create table WORKS_ON( Essn INT, Hours FLOAT, PRIMARY KEY(Essn), FOREIGN
KEY Essn REFERENCE Employee (Ssn));

Insert <999345,25.8> into WORKS_ON
```

Referential integrity Violation

Violations when DELETE

- DELETE may violate only referential integrity:
- If the primary key value of the tuple being deleted is referenced by other tuples in the database.
- Can be remedied by several actions: RESTRICT, CASCADE, SET NULL one of the above options must be specified for each foreign key constraint.

Example

```
CREATE TABLE Employee (
NIC           VARCHAR (10) PRIMARY KEY,
name          VARCHAR (50),
Works_in      INT,
CONSTRAINT fk_EmpDept FOREIGN KEY,
(works_in) REFERENCES Department(Dept_Nmbr),
ON DELETE CASCADE,
ON UPDATE NO ACTION
)
```

When DROP TABLE

- The actions to take when Dropping tables.
- RESTRICT- if there is constraint (FK/View) then do not drop the table.
- CASCADE – drop all the other constraints & views that refers the table.

DROP TABLE Employee [RESTRICT | CASCADE]

Add or Remove Constraints

- Drop a table's primary key constraint

```
Alter Table Student Drop Primary Key
```

- Drop a unique, foreign, or check constraint

```
Alter Table Employee Drop Constraint fk_EmpDept
```

- Add a new constraint

```
Alter Table PassStudents Add Constraint avg_Marks Check  (marks >= 50)
```

Stored Procedures

- A stored procedure is a prepared SQL code that you can save, so the code can be reused over and over again.
- If you have an SQL query that you write over and over again, save it as a stored procedure, and then just call it to execute it.
- You can also pass parameters to a stored procedure.

Stored Procedure Syntax

```
CREATE PROCEDURE procedure_name
AS
sql_statement
GO;
```

`EXEC procedure_name;`

```
CREATE PROCEDURE SelectAllCustomers
AS
SELECT * FROM Customers
GO;
```

`EXEC SelectAllCustomers;`

Stored Procedure With One Parameter

```
CREATE PROCEDURE SelectAllCustomers @City nvarchar(30)
AS
SELECT * FROM Customers WHERE City = @City
GO;

EXEC SelectAllCustomers @City = 'London';
```