# CSCI 4710/5710 Databases

Constraints

# CONSTRAINTS

- Constraints: rules used to enforce business rules, practices and policies.
- Ensure data accuracy and integrity.
- The row can’t be inserted if it violates any constraints.

# CONSTRAINTS

| CONSTRAINT | DESCRIPTION |
|---|---|
| PRIMARY KEY | Determines the column that is going to be the unique identifier for the records in that table. Can’t be null and the data values must be unique |
| FOREIGN KEY | Represents the relationship in the cases of: one-to-many relationship and one-to-one relationship. Ensures that if a value is enters in the specified column, it must already exist in the other table that it is referencing. |
| UNIQUE | Ensures that all data values in that specific column are unique. The difference between this and the primary key is that this allows null values |
| CHECK | Ensures that the values are correct before entering them, e.g. the ship date can’t be before the order date. |
| NOT NULL | Enforces the required attributes. |

# Creating Constraints

- When can you create constraints?
  - During table creation.
  - For an existing table: using ALTER TABLE keyword.

- Constraints can be applied at:
  - Column level.
  - Table level.

# PRIMARY KEY Constraint

- Ensures that columns do not contain duplicate or NULL values
- Can be added at table creation using the constraint keyword.
- Can be added to an existing table using the ADD constraint command.

COMPUTER_BOOKS

| Field | Type | Null | Key | Default | Extra |
|---|---|---|---|---|---|
| ISBN | varchar(10) | NO | | NULL | |
| Title | varchar(30) | YES | | NULL | |
| PubDate | date | YES | | NULL | |
| PubID | int(11) | YES | | NULL | |
| Cost | double(5,2) | YES | | NULL | |
| Retail | double(5,2) | YES | | NULL | |
| Discount | double(4,2) | YES | | NULL | |
| Category | varchar(12) | YES | | NULL | |

```
1  ALTER TABLE COMPUTER_BOOKS
2  ADD CONSTRAINT comp_boos_PK
3  PRIMARY KEY(ISBN);
4  describe COMPUTER_BOOKS
```

250% 1:3

Result Grid | Filter Rows: Search | Export:

| Field | Type | Null | Key | Default | Extra |
|---|---|---|---|---|---|
| ISBN | varchar(10) | NO | PRI | NULL | |
| Title | varchar(30) | YES | | NULL | |
| PubDate | date | YES | | NULL | |
| PubID | int(11) | YES | | NULL | |
| Cost | double(5,2) | YES | | NULL | |
| Retail | double(5,2) | YES | | NULL | |
| Discount | double(4,2) | YES | | NULL | |
| Category | varchar(12) | YES | | NULL | |

# PRIMARY KEY Constraint

- For composite primary key, include the parts of the key in the parenthesis and separate them by comma .

COMPUTER_BOOKS

| Field | Type | Null | Key | Default | Extra |
|---|---|---|---|---|---|
| ISBN | varchar(10) | NO | | NULL | |
| Title | varchar(30) | YES | | NULL | |
| PubDate | date | YES | | NULL | |
| PubID | int(11) | YES | | NULL | |
| Cost | double(5,2) | YES | | NULL | |
| Retail | double(5,2) | YES | | NULL | |
| Discount | double(4,2) | YES | | NULL | |
| Category | varchar(12) | YES | | NULL | |

```
1 • ALTER TABLE COMPUTER_BOOKS
2   ADD CONSTRAINT comp_boos_PK
3   PRIMARY KEY(ISBN , Title);
4 • describe COMPUTER_BOOKS
```

250% 25:3

Result Grid | Filter Rows: Search | Export:

| Field | Type | Null | Key | Default | Extra |
|---|---|---|---|---|---|
| ISBN | varchar(10) | NO | PRI | NULL | |
| Title | varchar(30) | NO | PRI | NULL | |
| PubDate | date | YES | | NULL | |
| PubID | int(11) | YES | | NULL | |
| Cost | double(5,2) | YES | | NULL | |
| Retail | double(5,2) | YES | | NULL | |
| Discount | double(4,2) | YES | | NULL | |
| Category | varchar(12) | YES | | NULL | |

# FOREIGN KEY Constraint

- Enforces referential integrity (a value to exist in the referenced column of another table).
- NULL values are allowed.
- Maps to the PRIMARY KEY in parent table.

```
1 • ALTER TABLE DEPENDENT ADD CONSTRAINT dep_empID_fk FOREIGN KEY (EmployeeID)
2   REFERENCES EMPLOYEE (employeeID);
3 • describe Dependent
```

250% 1:1

Result Grid | Filter Rows: Search | Export:

| Field | Type | Null | Key | Default | Extra |
|---|---|---|---|---|---|
| DependentID | int(11) | NO | PRI | NULL | |
| DependentName | varchar(255) | YES | | NULL | |
| EmployeeID | int(11) | YES | MUL | NULL | |
| Relationship | varchar(255) | YES | | NULL | |

# FOREIGN KEY Constraint

- Trying to insert data in the child table for a non-existing record in the parent table will result in an error.

```
1 • insert into dependent values (2323,'Jhonathan',5505,'son')
```

```
47  16:27:42  insert into dependent values (2323,'Jhonathan',5505,'son')
Error Code: 1452. Cannot add or update a child row: a foreign key constraint fails (`emp`.`dependent`, CONSTRAINT `dep_empID_fk` FOREIGN KEY (`EmployeeID`) REFERENCES `EMPLOYEE` (`EmployeeID`))
```

- Foreign Key will not allow deleting data from the parent table if that record has child/ren in another table

# FOREIGN KEY Constraint

- Using ON DELET CASCADE command will delete all children when you delete the parent.
- Dependent Table

| DependentID | DependentName | EmployeeID | Relationship |
|---|---|---|---|
| 1 | Sarah | 1001 | Wife |
| 2 | Mariah | 1001 | Child |
| 3 | Tim | 1001 | Child |
| 4 | Nate | 1006 | Husband |
| NULL | NULL | NULL | NULL |

```
1 • delete from employee where employeeid = 1001;
2 • select * from dependent;
3
4
```

250% 46:1

Result Grid | Filter Rows: Search | Edit: | Export/Import:

| DependentID | DependentName | EmployeeID | Relationship |
|---|---|---|---|
| 4 | Nate | 1006 | Husband |
| NULL | NULL | NULL | NULL |

# UNIQUE Constraint

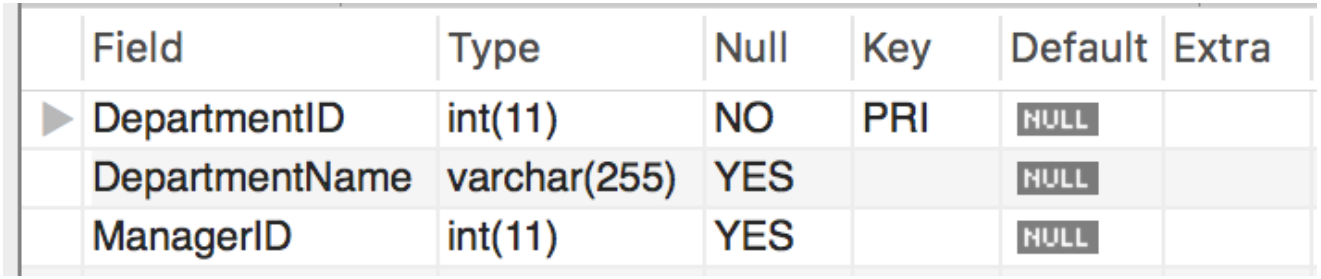
- UNIQUE constraints means that no duplications will be allowed in the referenced column.
  - Null values will still be allowed for that column.
  - Department table:

```
1  ALTER TABLE DEPARTMENT ADD CONSTRAINT
2  dep_name_unique UNIQUE (DepartmentName);
3  DESCRIBE DEPARTMENT;
4
```

250% 1:2

Result Grid | Filter Rows: Search | Export:

| Field | Type | Null | Key | Default | Extra |
|---|---|---|---|---|---|
| DepartmentID | int(11) | NO | PRI | NULL | |
| DepartmentName | varchar(255) | YES | UNI | NULL | |
| ManagerID | int(11) | YES | | NULL | |



| Field | Type | Null | Key | Default | Extra |
|---|---|---|---|---|---|
| DepartmentID | int(11) | NO | PRI | NULL | |
| DepartmentName | varchar(255) | YES | | NULL | |
| ManagerID | int(11) | YES | | NULL | |

# CHECK Constraint

- Control or limit values in some columns.
- Making sure that the order date is before the shipping date:

```
ALTER TABLE  ORDERS ADD CONSTRAINT
orders_shipdate_check CHECK(OrderDate < ShipDate);
```

- Limiting the values in “Relationship” column of the dependent table to “spouse” and “child”

```
ALTER TABLE DEPENDENT ADD CONSTRAINT
CHECK (Relationship in('Spouse','Child'));
```

# NOT NULL Constraint

- Can be added using ALTER TABLE command.
  - NOT NULL constraint is added
  - General syntax:

```
ALTER TABLE table_name
CHANGE
   old_column_name
   new_column_name column_definition;
```

  - Example:

```
ALTER TABLE AUTHOR2
CHANGE
fullname fullname varchar(255) not null;
```

# NOT NULL Constraint

- Can be removed using ALTER TABLE command.
  - NOT NULL constraint is removed
  - General syntax:

```
ALTER TABLE table_name
CHANGE
    old_column_name
    new_column_name column_definition;
```

  - Example:

```
ALTER TABLE AUTHOR2
CHANGE
fullname fullname varchar(255) ;
```

# Table Constraints Lookup

- You can lookup all table constraint as follows (example on ‘orders’ table):

```
1 • SELECT * FROM INFORMATION_SCHEMA.TABLE_CONSTRAINTS
2   WHERE TABLE_NAME = 'ORDERS'
```

160% 1:2

Result Grid | Filter Rows: Search | Export:

| CONSTRAINT_CATALOG | CONSTRAINT_SCHEMA | CONSTRAINT_NAME | TABLE_SCHEMA | TABLE_NAME | CONSTRAINT_TYPE |
|---|---|---|---|---|---|
| def | CS3200 | PRIMARY | CS3200 | orders | PRIMARY KEY |
| def | CS3200 | orders_customerID_fk | CS3200 | orders | FOREIGN KEY |

# Dropping Table Constraints

- You must know what constraint you need to drop first.
  - UNIQUE Constraint:

```
ALTER TABLE  TableName DROP INDEX UniqueConstraintName;
```

  - PRIMARY KEY Constraint:

```
ALTER TABLE  TABLENAME DROP PRIMARY KEY;
```

  - FOREIGN KEY Constraint:

```
ALTER TABLE  TABLENAME DROP FOREIGN KEY ForeignKeyConstraintName;
```

  - CHECK Constraint:

```
ALTER TABLE  TABLENAME DROP CHECK   CheckConstraintName   ;
```