



# Relational Database Concepts

*Integrity Rules Syntax*

PRESENTED BY IMGARD EKOKOBE



# Objectives

- Understand the key components of a relational database.
- Identify one to many relationships in associated primary keys and foreign keys in a relational database diagram.
- Identify rows with either missing foreign key values or orphan foreign key values.
- Identify foreign keys that are required, that is must not have a missing value and foreign keys that can have missing values.



# Why are constraint names important?

- Imagine that you are on call as a database administrator. An error occurs but the error name is meaningless.
- How would you troubleshoot the issue?



# Integrity Rules Syntax

The create table statement allows five different types of constraints. Primary key, foreign key, unique, required, and check constraints.

- **Unique constraints** indicate that a column, or combination of columns, as unique, that is no duplicate values for a table.
- **Unique constraints** are not specified for primary keys. As the primary key constraint already implies uniqueness.
- **Unique constraints** are specified for candidate keys. These are other columns besides the primary key, that are also unique on a table. For example, student table may have STDNo, as the primary key. An StdEmail as the candidate key can be specified with a unique constraint.
- **Required constraints** using the NOT NULL keywords indicate that a value must be entered for a column. This means that the column cannot be left blank.
- **Check constraints** specify conditions that a column must satisfy such as limits on a column containing a year value.

# Integrity Rules Syntax

Constraints may be placed in the CREATE TABLE statements:

**Inline** – as part of a column definition. For example in the Create Table DDL statement to create the Student Table in the university database below, the NOT NULL constraint is an INLINE constraint

```
CREATE TABLE Student
(stdNo          char(11) not null,
stdFirstName    varchar2(30) not null,
stdLastName     varchar2(30) not null,
stdCity         varchar2(30) not null,
stdState        char(2) not null,
stdZip          char(10) not null,
stdMajor        char(6),
stdClass        char(2),
stdGPA          decimal(3,2),
CONSTRAINT StudentPk PRIMARY KEY (StdNo) );
```



# Integrity Rules Syntax

**External** – after column definitions.

Let's take a look again at the Create Table DDL statement to create the Student Table in the university database below,

PRIMARY KEY constraint StudentPk constraint is an EXTERNAL constraint.

```
CREATE TABLE Student
(stdNo          char(11) not null,
stdFirstName    varchar2(30) not null,
stdLastName     varchar2(30) not null,
stdCity         varchar2(30) not null,
stdState        char(2) not null,
stdZip          char(10) not null,
stdMajor        char(6),
stdClass        char(2),
stdGPA          decimal(3,2),CONSTRAINT StudentPk PRIMARY KEY (StdNo) );
```



# Integrity Rules Syntax

- Constraints involving more than one column must be placed external to column definitions, as shown in the DDL for the Enrollment table below.

```
CREATE TABLE Enrollment
(OfferNo INTEGER not null,
StdNo char(11) not null,
EnrGrade decimal(3,2),
CONSTRAINT EnrollmentPK PRIMARY KEY (OfferNo, StdNo),
CONSTRAINT OfferingFK FOREIGN KEY (OfferNo) REFERENCES OfferingON DELETE
CASCADE,
CONSTRAINT StudentFK FOREIGN KEY (StdNo) REFERENCES student ON DELETE
CASCADE );
```



# Integrity Rules Syntax

- Let's review the syntax for writing constraints in the diagram below by isolating the constraint part of the statement
- In the first examples the PRIMARY KEY constraints is defined on the Course and Enrollment tables.

## Syntax

CONSTRAINT keyword constraint\_name constraint\_type (column\_name(s))

CONSTRAINT PKCourse PRIMARY KEY (CourseNo)

CONSTRAINT PKCourse PRIMARY KEY (CourseNo)

CONSTRAINT PKErollment PRIMARY KEY (OfferNo, StdNo)

CONSTRAINT UniqueCrsDesc UNIQUE (CrsDesc)

CONSTRAINT FKOfferNo FOREIGN KEY (OfferNo)  
REFERENCES Offering

CONSTRAINT OffCourseNoReq NOT NULL





# Integrity Rules Syntax

In the second example the PRIMARY KEY constraints is defined externally on the Enrollment table.

The constraint is placed externally, because the primary key is the combination of two columns(OfferNo and StdNo).

Syntax

CONSTRAINT keyword constraint\_name constraint\_type (column\_name(s))

CONSTRAINT PKErollment PRIMARY KEY (StdNo, OfferNo)



# Integrity Rules Syntax

The third example shows the syntax for defining a unique constraint with a unique keyword.

The constraint named UniqueCrs description defines a unique constraint for the Crs description column

Syntax

```
CONSTRAINT keyword constraint_name constraint_type (column_name(s))
```

```
CONSTRAINT UniqueCrsDesc UNIQUE (CrsDesc)
```



# Integrity Rules Syntax

The fourth example defines a foreign key constraint with a foreign key column that is offer number, and parent table offering specified in the constraint.

Syntax

```
CONSTRAINT keyword constraint_name constraint_type (column_name(s))  
CONSTRAINT FKOfferNo FOREIGN KEY (OfferNo) REFERNECES Offering
```



# Integrity Rules Syntax

The fifth example defines a required constraint with the NOT NULL keywords. Required constraints are usually specified inline as they apply to individual columns, not combination of columns.

## Syntax

```
CONSTRAINT keyword constraint_name constraint_type (column_name(s))  
CONSTRAINT OfferCourseNoReq NOT NULL
```



# Integrity Rules Syntax

- The constraint name is optional but good practice to identify the error when a constraint violation occurs.
- The comma is necessary after the inline constraint to terminate the column definition.
- The keyword CONSTRAINT keyword is optional for inline constraints, but required for external constraints.

For example, it is omitted in the CREATE TABLE statements to create the table for the Enrollment table:

```
CREATE TABLE Enrollment  
(OfferNo INTEGER not null,  
StdNo char(11) not null,  
EnrGrade decimal(3,2),  
CONSTRAINT EnrollmentPK PRIMARY KEY (OfferNo, StdNo),  
CONSTRAINT OfferingFK FOREIGN KEY (OfferNo) REFERENCESOffering ON DELETE CASCADE,  
CONSTRAINT StudentFK FOREIGN KEY (StdNo) REFERENCESStudent ON DELETE CASCADE );
```



# Integrity Rules Syntax

- It is good practice to use constraint names to facilitate problem solving when a constraint violation occurs.
- A DBA on call will be helped at least initially by seeing a meaningful constraint name when a violation occurs such as a non-unique primary key value.