**Theory**

**REFERENTIAL INTEGRITY**

Referential Integrity is set of constraints applied to foreign key which prevents entering a row in child table (where you have foreign key) for which you don't have any corresponding row in parent table i.e. entering NULL or invalid foreign keys. *Referential Integrity* prevents your table from having  incorrect or incomplete relationship 

The constraints available in SQL are [**Foreign Key**](https://beginner-sql-tutorial.com/sql-integrity-constraints.htm#sql-foreign-constraint)**,**[**Not Null**](https://beginner-sql-tutorial.com/sql-integrity-constraints.htm#sql-not-null-constraint)**,**[**Unique**](https://beginner-sql-tutorial.com/sql-integrity-constraints.htm#sql-unique-constraint)**,**[**Check**](https://beginner-sql-tutorial.com/sql-integrity-constraints.htm#sql-check-constraint)**.**

Constraints can be defined in two ways   
1) The constraints can be specified immediately after the column definition. This is called column-level definition.   
2) The constraints can be specified after all the columns are defined. This is called table-level definition.

**Foreign Key**

A FOREIGN KEY is a key used to link two tables together. A FOREIGN KEY is a field (or collection of fields) in one table that refers to the PRIMARY KEY in another table.The table containing the foreign key is called the child table, and the table containing the candidate key is called the referenced or parent table.

Syntax

CREATE TABLE Orders (  
    OrderID int NOT NULL,  
    OrderNumber int NOT NULL,  
    PersonID int,  
    PRIMARY KEY (OrderID),  
    FOREIGN KEY (PersonID) REFERENCES Persons(PersonID)  
);

**Null Value**

A field with a NULL value is a field with no value.

If a field in a table is optional, it is possible to insert a new record or update a record without adding a value to this field. Then, the field will be saved with a NULL value.

A foreign key that whose columns omit [NOT NULL](https://www.vertica.com/docs/9.2.x/HTML/Content/Authoring/AdministratorsGuide/Constraints/ConstraintTypes/NotNULLConstraints.htm) can contain NULL values, even if the primary key contains no NULL values. Thus, you can insert rows into the table even if their foreign key is not yet known.

**Parent child table**

A foreign key is a way to enforce referential integrity within your Oracle database. A foreign key means that values in one table must also appear in another table.

The referenced table is called the parent table while the table with the foreign key is called the child table. The foreign key in the child table will generally reference a [primary key](https://www.techonthenet.com/oracle/primary_keys.php) in the parent table.

**Not NULL constraint**

By Default, the columns are able to hold NULL values. A NOT NULL constraint in SQL is used to prevent inserting NULL values into the specified column, considering it as a not accepted value for that column. This means that you should provide a valid SQL NOT NULL value to that column in the INSERT or UPDATE statements, as the column will always contain data.

CREATE TABLE Persons (  
    ID int NOT NULL,  
    LastName varchar(255) NOT NULL FOREIGN KEY (PersonID) REFERENCES Persons(PersonID),  
    FirstName varchar(255) NOT NULL,  
    Age int  
);

**Queries**

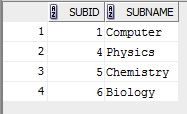
**1**

create table Subject(

subid int NOt NULL,

subname varchar(10) NOT NULL

);



insert into subject values(1,'rithin');

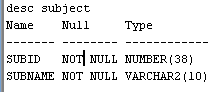
insert into subject values(2);

insert into subject values('roop');

insert into subject values(4,'romal');

insert into subject values(5,'Rohaan');

insert into subject values(6,'Richu');



ALTER TABLE Subject

MODIFY subid int primary key;



**2**

create table Staff(

staffid int constraint test4 UNIQUE,

staffname varchar(10),

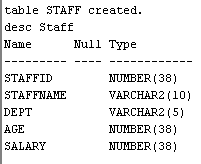
dept varchar(5),

age int constraint test2 check(age>22),

salary int constraint test3 check(salary<35000)

);

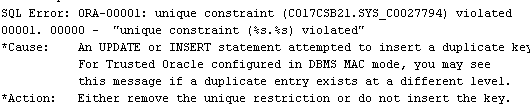
desc Staff;



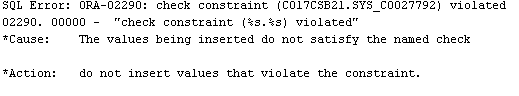
insert into staff values(1,'Richu','cs',23,20000);



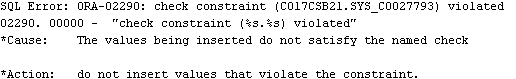
insert into staff values(1,'Rithin','cs',23],20000);



insert into staff values(2,'Rithin','cs',20,20000);



insert into staff values(2,'Rithin','cs',23,50000);



alter table staff

drop constraint test3;



alter table staff

drop constraint test4;



**3**

create table bank(

bankcode varchar(3),

bankname varchar(10) NOT NULL,

headoffice varchar(10),

branch int check(branch>0)

);



insert into bank values('EKM','r','banglore',4);



**4**

create table Branch(

branchid int primary key,

branchname varchar(10) default('New Delhi'),

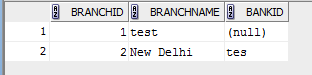
bankid varchar(3),

FOREIGN KEY (bankid) REFERENCES Bank(bankcode)

);



**insert into branch(branchid,bankid) values(2,'tes');**



5

Create view Sales\_staff as

select \* from staff where dept='sales';



**6**

replace view sales\_staff as

select \* from staff where dept='sales' and salary>20000;

7

drop table branch;

create table branch(

branchid int constraint PK primary key,

branch\_name varchar(20) default 'New Delhi',

bank\_id int,

constraint FK FOREIGN KEY (bank\_id) REFERENCES Bank(bankcode)

);





8

ALTER TABLE branch

DROP constraint pk;

ALTER TABLE branch

DROP constraint DEFAULT;





drop view sales\_staff;

