NIBM WORLD WIDE

SOFTWARE ENGINEERING

DATABASE MANAGEMENT SYSTEMS

DATA DEFINITION LANGUAGE (DDL)

National Institute of Business Management

Lesson 07 – Data Definition Language (DDL)

Data Definition Language (DDL)

These statements are used to create new objects, alter the structure of existing objects, or to remove objects from the system. These commands cannot use to modify data.

1. CREATE DATABASE

Syntax: Create Database DatabaseName;

Exercise: Create A Database Called Company

Answer: Create Database Company;

2. SELECT DATABASE

Syntax: Use DatabaseName;

Exercise: Select the Database Company

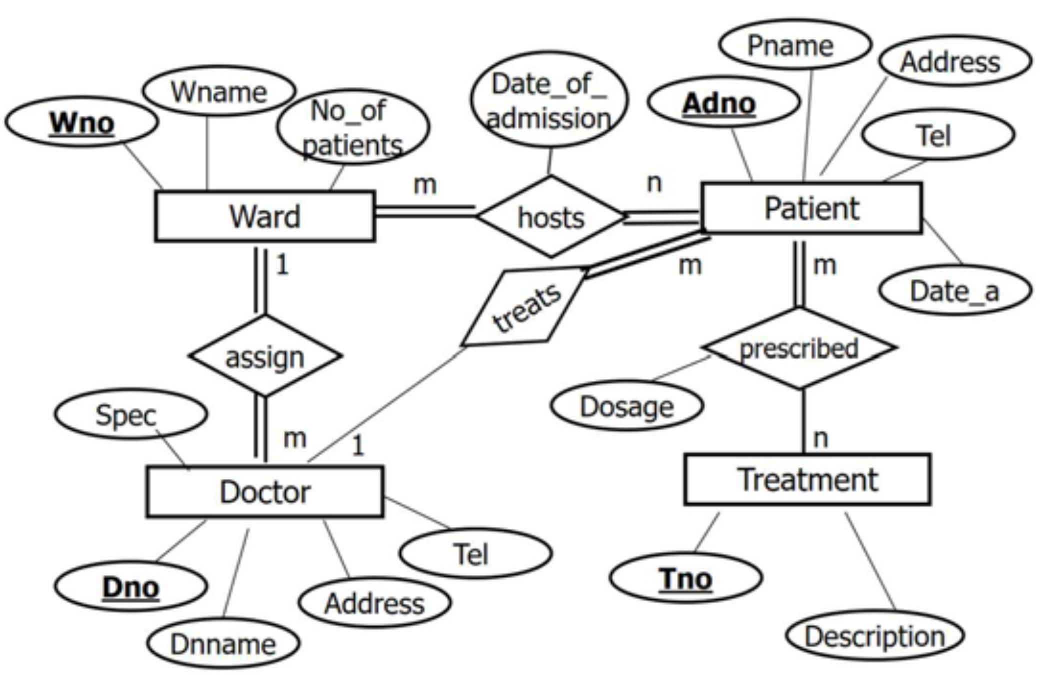
Answer: Use Company;

3. CREATE TABLE

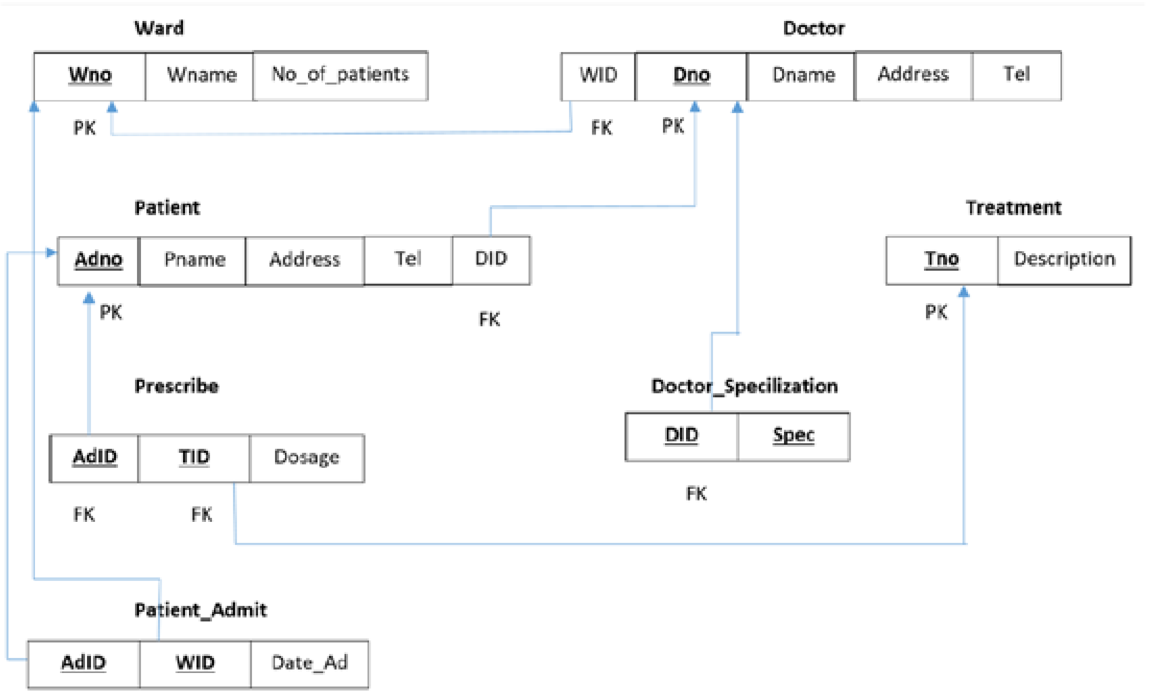
Before create tables in the Database we need to learn how to Design the Table First.

Let's take an example for File Design

ER/EER Diagrams



Schema Mapping



After Schema Mapping you have to Normalize the tables if there any issue in Schema Mapping.

File Design (Before Create Practical Tables in the DBMS Tools)

Ward

Primary Key: Wno

Field Names	Data type	Data Size	Description
Wno	Text	05	Ward Number
Wname	Text	30	Ward Name
No_of_Patients	Number	05	Capacity of the Ward
Record Size		40	

Doctor

Primary Key: Dno

Foreign Key: WID

Field Names	Data type	Data Size	Description
Dno	Text	05	Doctor Number
Dname	Text	30	Doctor Name
Address	Text	50	Doctor Address
Tel	Number	10	Doctor Telephone
WID	Text	05	Ward Number
Record Size	100		

Patient

Primary Key: Adno

Foreign Key: DID

Field Names	Data type	Data Size	Description
Adno	Text	05	Patient Number
Pname	Text	50	Patient Name
Address	Text	50	Patient Address
Tel	Number	10	Patient Telephone
DID	Text	05	Doctor Number
Record Size	120		

Treatment

Primary Key: Tno

Field Names	Data type	Data Size	Description
Tno	Text	10	Treatment Number
Description	Text	30	Treatment Description
Record Size	35		

Prescribe

Primary Keys: AdID, TID

Foreign Keys: AdID, TID

Field Names	Data type	Data Size	Description
AdID	Text	05	Patient Number
TID	Text	10	Treatment Number
Dosage	Text	10	Treatment Dosage
Record Size	20		

Doctor_Specilization

Primary Keys: DID, Spec

Foreign Key: DID

Field Names	Data type	Data Size	Description
DID	Text	05	Doctor Number
Spec	Text	30	Doctor Specialization
Record Size	35		

Patient_Admit

Primary Keys: AdID, WID

Foreign Keys: AdID, WID

Field Names	Data type	Data Size	Description
AdID	Text	05	Patient Number
WID	Text	05	Ward Number
Date_Ad	Date		Date of Admission
Record Size	10		

Example:

Client

Primary Key: Client_Id

Field Names	Data type	Data Size	Description
Client_Id	Text	10	Client Id
Client_Name	Text	20	Client Name
Client_Address	Text	20	Client Address
TP	Number	10	Client Telephone
Balance_Due	Number	10	Client Balance
Order_Date	Date		Client Order Date

Common data types

INT(n)

n = maximum number of digits

VARCHAR (n)

n= maximum length of character strings

CHAR (n)

n= Fix length of character strings. In Char data type, the size is fix. As an example if the size is 5 which means there must be 5 characters.

DECIMAL (n, d)

n = maximum number of digits

d = maximum number of digits right of the decimal

DATE

The standard format is YYYY-MM-DD

Syntax: Create Table table_name (column_name datatype(size), column_name datatype(size) …);

Exercise: Create a Table called Client

Answer: Create Table Client (Client_Id varchar(10), Client_Name varchar(20), Client_Address varchar(20), TP int, Balance_due int, Order_date Date, primary key(Client_Id));

INSERT DATA INTO TABLE

Since we need to do table constraints, let do the Insert operation first. **Remember this is a Data Manipulation Operation.**

Syntax: Insert into table_name (column_name, column_name….) values (expression, expression…);

Exercise: Insert following data

Client_Id	Client_Name	Client_Address	TP	Balance_due	Order_date
C100	Amali	Colombo	0772569047	5000	2012-05-11
C200	Saman	Kandy	0712569047	12500	2010-06-11

Table 7.0.1 Insert Data into the Client Table

Answer:

First Method

```
Insert into Client (Client_Id, Client_Name, Client_Address, TP, Balance_due, Order_date)
values ('C100', 'Amali', 'Col07',0772569014, 5000, '2012-02-13');
```

Second Method

```
Insert into Client values ('C100', 'Amali', 'Col07',0772569014, 5000, '2012-02-13');
```

Remember Second Method is only possible if the data is available for all the columns. Suppose C100 Client doesn’t have a Telephone number. Then you have to write the Insert command as follow.

```
Insert into Client (Client_Id, Client_Name, Client_Address, Balance_due, Order_date) values
('C100', 'Amali', 'Col07', 5000, '2012-02-13');
```

When you insert the next data remember to change the Client Id. Because Client Id is the Primary Key and Primary Key value should be unique.

4. CREATE TABLES WITH CONSTRAINTS

NULL values Concept

Setting a null value is appropriate when the actual value is unknown or when a value would not be meaningful. A null value is not equivalent to a value of zero.

When a column name is defined as not null, then that column becomes a mandatory column. It implies that the user is forced to enter data into that column.

Example:

Create table Client (Client_Id varchar(10), Client_Name varchar(20) not null, Client_Address varchar(20) not null, TP int not null, Balance_due int not null, Order_date Date not null, primary key(Client_Id));

Here you have to change the Client Table Name because inside the Company Database you have to give unique name for table. Now try to insert a Telephone number without a value, you will get an error.

Default values Concept

At the time of cell creation, a ‘default value’ can be assigned to it. The data type of the default value should match the data type of column.

Example:

Create table Client (Client_Id varchar(10), Client_Name varchar(20) not null, Client_Address varchar(20) not null default 'Colombo', TP int not null, Balance_due int not null, Order_date Date not null, primary key(Client_Id));

Check Integrity Constraints

Use CHECK constraint when you need to enforce integrity rule that can be evaluated based on a logical expression.
Following are few examples of appropriate CHECK constraint:

Example:

A CHECK constraint on the Client Id column of client so that Client Id values starts with ‘C’.

Create table Client (Client_Id varchar(10) constraint ck_id CHECK (Client_Id like 'C%'), Client_Name varchar(20) not null, Client_Address varchar(20) not null default 'Colombo', TP int not null, Balance_due int not null, Order_date Date not null, primary key(Client_Id));

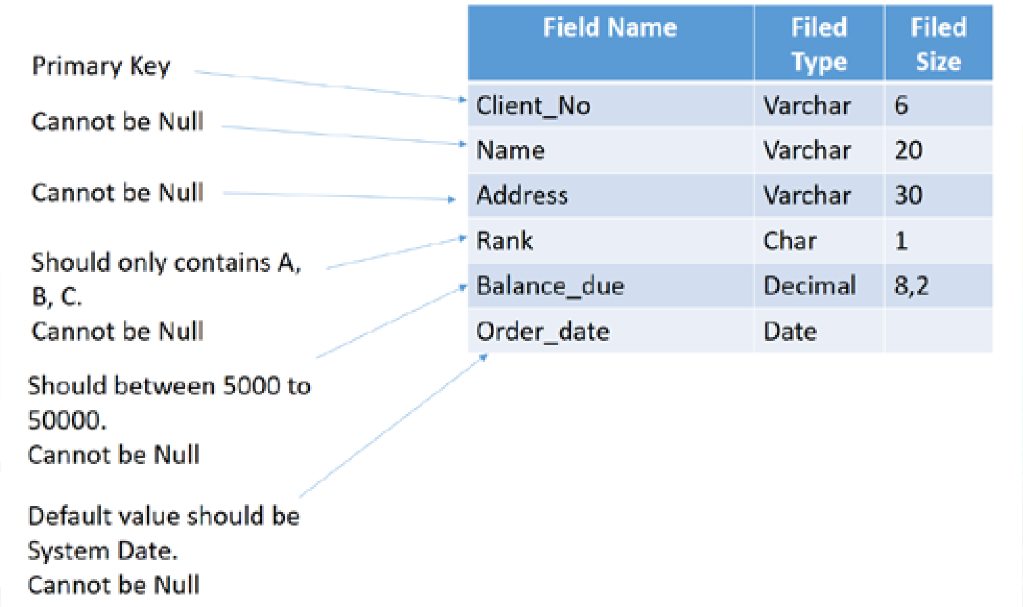
Example:

A CHECK constraint on the address column of client so that only the address “Colombo”, “Galle”, “Matara”, and “Jaffna” are allowed.

Create table Client (Client_Id varchar(10) primary key constraint c_id CHECK (Client_Id LIKE ('C%')), Client_Name varchar(20) not null, Client_Address varchar(20) not null default

```
'Colombo' constraint ck_address check (Client_Address IN ('Colombo', 'Galle', 'Jaffna')), TP
int not null, Balance_due int not null, Order_date Date not null);
```

Exercise:



Primary Key

Cannot be Null

Cannot be Null

Should only contains A, B, C.
Cannot be Null

Should between 5000 to 50000.
Cannot be Null

Default value should be System Date.
Cannot be Null

Field Name	Filed Type	Filed Size
Client_No	Varchar	6
Name	Varchar	20
Address	Varchar	30
Rank	Char	1
Balance_due	Decimal	8,2
Order_date	Date	

Answer:

```
Create table Client (Client_No varchar(6) primary key, Name varchar(20) not null, address
varchar (30) not null,  Rank varchar(1) check (rank in ('A', 'B', 'C')), balance_due
decimal(8,2) check (balance_due > 5000 and balance_due < 50000), order_date date default
getdate());
```

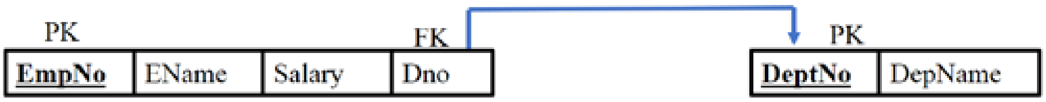
Referential Integrity Constraint (Foreign Key)

Foreign key provides referential integrity rules either within a table or among tables. A foreign key is used in a relationship with either a primary key of the same table or another table.

There are different ways to create a Foreign Key on another table with CREATE TABLE command with the FOREIGN KEY column constraint.

Example:

An employee works for a department. One employee belongs to one department and many employees will be there in a one department.



First create the Department table. Then create Employee table with a foreign key by taking primary key of Department table.

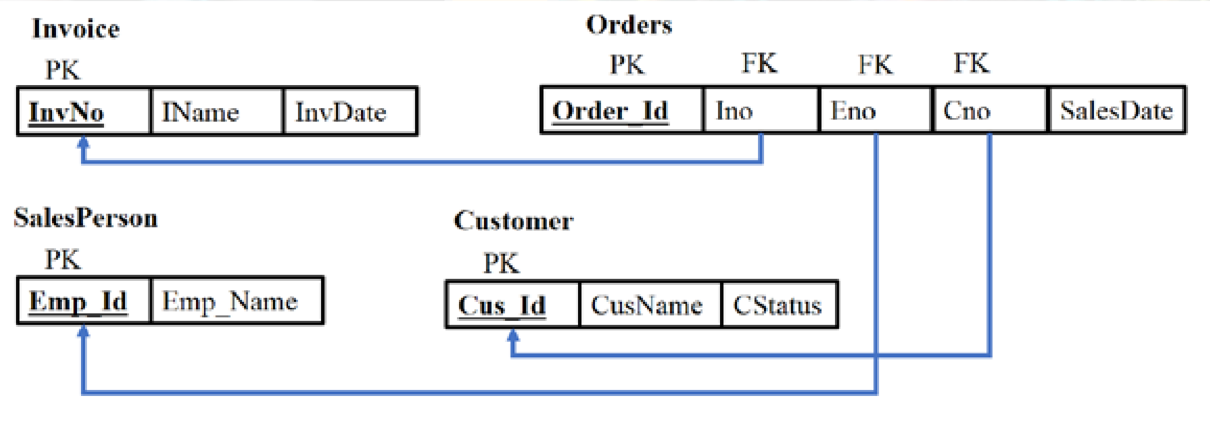
Create table Department (DeptNo varchar(3) primary key, DepName varchar(10));

Create table Employee (EmpNo varchar(3) primary key, EName varchar (10), Salary int, Dno varchar(3) References Department (DeptNo));

Another Way

Create table Employee (EmpNo varchar(4) primary key, EName varchar(10), Salary int, Dno varchar(5), Foreign key References Department (DeptNo));

Exercise:



Create table Invoice (InvNo varchar(3) Primary Key, IName varchar (5));

Create table SalesPerson (Emp_Id varchar(3) primary key, Emp_Name varchar(10));

Create table Customer (Cus_Id varchar(3) primary key, CusName varchar(10), C_Status varchar(5));

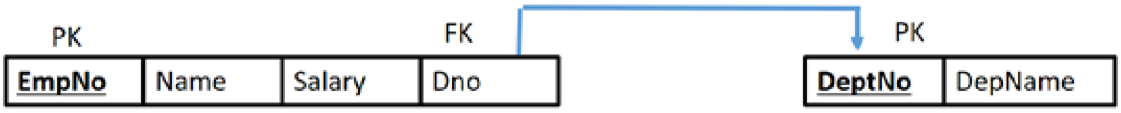
Create table Orders (Order_Id varchar(3), SalesDate date, Ino varchar(3), Eno varchar(3), Cno varchar(3), foreign key (Ino) references Invoice (InvNo), foreign key (Eno) references SalesPerson (Emp_Id), foreign key (Cno) references Customer (Cus_Id));

Composite Primary Key

If there are more than one primary key in the table, you can set composite primary keys.

Example:

Suppose EmpNo and Dno are primary keys of the Employee table.



Create table Employee (EmpNo varchar(4), Name varchar(10), Salary int, Dno varchar(5) Foreign key References Department (Dno), Primary Key (EmpNo, Dno));

5. ALTER TABLE

Add Field

Syntax:

Alter table table_name add newcolumn_name data_type(size);

Exercise:

Add the following columns to the table client.

Column Name	Data Type
Fax No	int
Amount	int

Table 7.0.2 Add new Columns to the Table

Answer:

Alter table Client add Fax_No int, Amount int;

Change Field Type/Field Size

Syntax:

Alter table table_name alter column column_name newdata_type(size), ...;

Exercise:

Change Datatype of Fax_no Field as Varchar

Alter table Client alter column Fax_no varchar(8);

Drop Field

Syntax:

Alter table drop column table_name column_name;

Exercise:

Drop Field Fax_no

Alter table Client drop column Fax_no;

Foreign Key On Alter Table

Syntax:

Alter Table table_name add foreign key column_name references table_name (column_name);

To create a FOREIGN KEY constraint on the “DeptNo" column when the “Employee" table is already created, use the following command. But remember before you execute this command first you should add DeptNo column to the Employee table.

Exercise:

Alter Table employee add foreign key (deptno) references department (deptno);

Another Way

Alter Table employee add constraint fk_employee_dept foreign key (deptno) references department (deptno);

Drop the Foreign Key Constraint

Syntax:

Alter Table table_name drop constraint constraint_name

Exercise:

Alter Table employee drop constraint fk_employee_dept

6. DROP TABLE

Syntax:

```
Drop Table table_name
```

If you have data in the table, it will delete permanently. The data and Table structure both will remove from the database. Once the Table is deleted, you cannot get it back.

Exercise:

```
Drop table Client;
```

7. TRUNCATE TABLE

Syntax:

```
Truncate Table table_name
```

This statement will remove all the data from the table but table structure will remain as it is.

Example:

```
Truncate Table Client;
```

8. RENAME TABLE

SQL Server does not have any statement that directly renames a table. However, it does provide you with a stored procedure named sp_rename that allows you to change the name of a table. Stored Procedures we will cover later in this E-Book.

Syntax:

```
EXEC sp_rename 'old_table_name', 'new_table_name'
```

Example:

```
EXEC sp_rename 'Client', 'Client_Master'
```

9. DESCRIBE TABLE

The simplest way to use a stored procedure sp_columns to show the columns and related information about a SQL Server table is to execute the stored procedure passing it the table name.

Syntax:

```
Exec sp_columns table_name;
```

Example:

```
Exec sp_columns Client;
```

10. DROP THE DATABASE

You can drop the Database using the following syntax. Remember before you Drop the Database, first of all you must select a different database.

Syntax:

```
Drop Database database_name;
```

Example:

```
Drop Database Company;
```