**SQL (MSBI)**

**SQL Server database can be created, altered and dropped**  
1. Graphically using SQL Server Management Studio (SSMS) or  
2. Using a Query

**To create the database graphically**  
1. Right Click on Databases folder in the Object explorer  
2. Select New Database  
3. In the New Database dialog box, enter the Database name and click OK.

**Creating** **Database Using Query**

Create Database DatabaseName

**To alter a database, once it's created**  
Alter database DatabaseName Modify Name = NewDatabaseName

**To Delete or Drop a database**  
Drop Database DatabaseThatYouWantToDrop

To create tblPerson and tblGender tables and establish primary key and foreign key constraints.

Create Table **tblGender**  
(ID int Not Null Primary Key,  
Gender nvarchar(50))

**To add a foreign key reference using a query**

Alter table tblPerson   
add constraint tblPerson\_GenderId\_FK FOREIGN KEY (GenderId) referencestblGender(ID)  
  
  
**The general formula is here**  
Alter table ForeignKeyTable add constraintForeignKeyTable\_ForiegnKeyColumn\_FK   
FOREIGN KEY (ForiegnKeyColumn) references PrimaryKeyTable (PrimaryKeyColumn)

**Altering an existing column to add a default constraint:**  
ALTER TABLE { TABLE\_NAME }  
ADD CONSTRAINT { CONSTRAINT\_NAME }  
DEFAULT { DEFAULT\_VALUE } FOR { EXISTING\_COLUMN\_NAME }  
  
**Adding a new column, with default value, to an existing table:**  
ALTER TABLE { TABLE\_NAME }   
ADD { COLUMN\_NAME } { DATA\_TYPE } { NULL | NOT NULL }   
CONSTRAINT { CONSTRAINT\_NAME } DEFAULT { DEFAULT\_VALUE }  
  
  
**The following command will add a default constraint, DF\_tblPerson\_GenderId.**  
ALTER TABLE tblPerson  
ADD CONSTRAINT DF\_tblPerson\_GenderId  
DEFAULT 1 FOR GenderId

**To drop a constraint**  
ALTER TABLE { TABLE\_NAME }   
DROP CONSTRAINT { CONSTRAINT\_NAME }

**The following check constraint, limits the age between ZERO and 150.**  
ALTER TABLE tblPerson  
ADD CONSTRAINT CK\_tblPerson\_Age CHECK (Age > 0 AND Age < 150)  
  
  
**The general formula for adding check constraint in SQL Server:**  
ALTER TABLE { TABLE\_NAME }  
ADD CONSTRAINT { CONSTRAINT\_NAME } CHECK ( BOOLEAN\_EXPRESSION )

**To drop the CHECK constraint:**  
ALTER TABLE tblPerson  
DROP CONSTRAINT CK\_tblPerson\_Age

**Basic select statement syntax**  
SELECT Column\_List  
FROM Table\_Name  
  
**If you want to select all the columns, you can also use \*. For better performance use the column list, instead of using \*.**  
SELECT \*  
FROM Table\_Name

**To Select distinct rows use DISTINCT keyword**  
SELECT DISTINCT Column\_List  
FROM Table\_Name  
  
**Example**: Select distinct city from tblPerson  
  
**Filtering rows with WHERE clause**  
SELECT Column\_List  
FROM Table\_Name  
WHERE Filter\_Condition  
  
**Example:** Select Name, Email from tblPerson where City = 'London'

Filtering rows using WHERE clause, before aggrgations take place:  
**Select City, SUM(Salary) as TotalSalary**  
**from tblEmployee**  
**Where City = 'London'**  
**group by City**  
  
Filtering groups using HAVING clause, after all aggrgations take place:  
**Select City, SUM(Salary) as TotalSalary**  
**from tblEmployee**  
**group by City**  
**Having City = 'London'**

**It is also possible to combine WHERE and HAVING**  
Select City, SUM(Salary) as TotalSalary  
from tblEmployee  
Where Gender = 'Male'  
group by City  
Having City = 'London'

**SQL Script to create tblEmployee and tblDepartment tables**

Create table tblDepartment

(

     ID int primary key,

     DepartmentName nvarchar(50),

     Location nvarchar(50),

     DepartmentHead nvarchar(50)

)

Go

Insert into tblDepartment values (1, 'IT', 'London', 'Rick')

Insert into tblDepartment values (2, 'Payroll', 'Delhi', 'Ron')

Insert into tblDepartment values (3, 'HR', 'New York', 'Christie')

Insert into tblDepartment values (4, 'Other Department', 'Sydney', 'Cindrella')

Go

Create table tblEmployee

(

     ID int primary key,

     Name nvarchar(50),

     Gender nvarchar(50),

     Salary int,

     DepartmentId int foreign key references tblDepartment(Id)

)

Go

Insert into tblEmployee values (1, 'Tom', 'Male', 4000, 1)

Insert into tblEmployee values (2, 'Pam', 'Female', 3000, 3)

Insert into tblEmployee values (3, 'John', 'Male', 3500, 1)

Insert into tblEmployee values (4, 'Sam', 'Male', 4500, 2)

Insert into tblEmployee values (5, 'Todd', 'Male', 2800, 2)

Insert into tblEmployee values (6, 'Ben', 'Male', 7000, 1)

Insert into tblEmployee values (7, 'Sara', 'Female', 4800, 3)

Insert into tblEmployee

values (8, 'Valarie', 'Female', 5500, 1)

Insert into tblEmployee values (9, 'James', 'Male', 6500, NULL)

Insert into tblEmployee values (10, 'Russell', 'Male', 8800, NULL)

Go

**General Formula for Joins**  
SELECT      ColumnList  
FROM           LeftTableName  
JOIN\_TYPE  RightTableName  
ON                 JoinCondition

**Join Or Inner Join**

SELECT Name, Gender, Salary, DepartmentName  
FROM tblEmployee  
INNER JOIN tblDepartment  
ON tblEmployee.DepartmentId = tblDepartment.Id  
  
**OR**  
  
SELECT Name, Gender, Salary, DepartmentName  
FROM tblEmployee  
JOIN tblDepartment  
ON tblEmployee.DepartmentId = tblDepartment.Id

**Left Join**

SELECT Name, Gender, Salary, DepartmentName  
FROM tblEmployee  
LEFT OUTER JOIN tblDepartment  
ON tblEmployee.DepartmentId = tblDepartment.Id  
  
**OR**  
  
SELECT Name, Gender, Salary, DepartmentName  
FROM tblEmployee  
LEFT JOIN tblDepartment  
ON tblEmployee.DepartmentId = tblDepartment.Id

**Right Join**

SELECT Name, Gender, Salary, DepartmentName  
FROM tblEmployee  
RIGHT OUTER JOIN tblDepartment  
ON tblEmployee.DepartmentId = tblDepartment.Id  
  
**OR**  
  
SELECT Name, Gender, Salary, DepartmentName  
FROM tblEmployee  
RIGHT JOIN tblDepartment  
ON tblEmployee.DepartmentId = tblDepartment.Id

**Full Join**

SELECT Name, Gender, Salary, DepartmentName  
FROM tblEmployee  
FULL OUTER JOIN tblDepartment  
ON tblEmployee.DepartmentId = tblDepartment.Id  
  
OR  
  
SELECT Name, Gender, Salary, DepartmentName  
FROM tblEmployee  
FULL JOIN tblDepartment  
ON tblEmployee.DepartmentId = tblDepartment.Id

**Query:**  
SELECT       Name, Gender, Salary, DepartmentName  
FROM           tblEmployee E  
LEFT JOIN   tblDepartment D  
ON                 E.DepartmentId = D.Id  
WHERE        D.Id IS NULL

**Query:**  
SELECT         Name, Gender, Salary, DepartmentName  
FROM             tblEmployee E  
RIGHT JOIN    tblDepartment D  
ON                   E.DepartmentId = D.Id  
WHERE          E.DepartmentId IS NULL

**Self Join**

SelectE.Name as Employee, M.Name as Manager  
fromtblEmployee E  
Left JointblEmployee M  
OnE.ManagerId = M.EmployeeId

**Inner Self Join tblEmployee table:**  
SelectE.Name as Employee, M.Name as Manager  
fromtblEmployee E  
Inner JointblEmployee M  
OnE.ManagerId = M.EmployeeId  
  
**Cross Self Join tblEmployee table:**  
SelectE.Name as Employee, M.Name as Manager  
fromtblEmployee  
Cross JointblEmployee