SQL Concepts

1) **Joins**

```
select c.LastName,c.FirstName,e.EmployeeId,d.Name as Department,r.Rate
from Contact c
join Employee e on c.ContactID = e.ContactID
join EmployeeDepartmentHistory h on e.EmployeeId = h.EmployeeId
join Department d on d.DepartmentId = h.DepartmentId
join EmployeePayHistory r on r.EmployeeId = e.EmployeeId
where c.FirstName like 'P%'
```

Refer : https://www.w3schools.com/sql/sql_join.asp

2) **Aggregate functions – Group by clause**

```
SELECT year(RateChangeDate) as YEAR , Count(Rate) as 'NO OF RATES CHANGED',
COUNT (DISTINCT Rate) as DISTINCTRATES
from EmployeePayHistory
where year(RateChangeDate) = 1996 OR
year(RateChangeDate) = 1997 OR
year(RateChangeDate) = 1998
GROUP by year(RateChangeDate)
```

Refer : https://www.w3schools.com/sql/sql_count_avg_sum.asp

3) **Stored Procedure**

```
//*CREATE PROCEDURE [GetEmployeeDetails]
@StateProvinceCode nvarchar(3)
AS *// Template
SELECT c.Lastname,c.firstname,a.city,s.StateProvinceCode
from Contact c
join Employee e on c.ContactId = e.ContactId
join EmployeeAddress ea on e.EmployeeID = ea.EmployeeID
join Address a on ea.AddressID = a.AddressID
```

join StateProvince s on a.StateProvinceId = s.StateProvinceId

where s.StateProvinceCode = @StateProvinceCode

(Note : To execute the stored procedure to verify the sample output

Code : Exec <Procedure name> @<variablename>;

i.e., Exec GetEmployeeDetails @ StateProvinceCode = 'AB'.

* donot submit the exec statement.

If procedure name once created we can't modify the code To modify please drop the procedure first and then execute the code.

Syntax to drop the procedure name : Drop <Procedurename>

Refer : https://www.w3schools.com/sql/sql_stored_procedures.asp

4) Joins

select CONCAT(c.FirstName,' ',c.LastName) as Name ,d.Name as Department, (SELECT CONCAT(c.FirstName,' ',c.LastName)

from Contact c

join Employee e on e.ContactID = c.ContactID

where e.EmployeeId = e.ManagerId

) as Manager,

a.City,s.StateProvinceCode,s.CountryRegionCode

from Contact c

join Employee e on e.ContactID = c.ContactID

join EmployeeDepartmentHistory h on h.EmployeeId = e.EmployeeId

join Department d on d.DepartmentId = h.DepartmentId

join EmployeeAddress ea on ea.EmployeeId = e.EmployeeId

join Address a on a.AddressId = ea.AddressId

join StateProvince s on s.StateProvinceId = a.StateProvinceId

where a.City = 'Phoenix'

order by d.Name

5) Pagination

```
SELECT Employeeid,Managerid,Title,Gender
 FROM Employee
  ORDER BY Employeeid
  OFFSET 5 ROWS
  FETCH NEXT 3 ROWS ONLY;
```

Refer : https://www.dofactory.com/sql/order-by-offset-fetch

6) Function

//*CREATE Function [----]

Return int

AS *// Template

Code : select statement ---(using joins) with where condition.

To view output call the function.

Please refer the link : https://stackoverflow.com/questions/21358306/how-to-execute-function-in-sql-server-2008

Note : Donot submit the code with execute statement finally.

7) Ranking

```
SELECT year(ModifiedDate) as MODIFIEDYEAR, Rate
   ,DENSE_RANK() OVER
   (PARTITION BY year(ModifiedDate) ORDER BY Rate DESC) AS RATERANKING
FROM EmployeePayHistory
WHERE year(ModifiedDate) > '2000'
ORDER BY year(ModifiedDate);
```

Refer : http://www.besttechtools.com/articles/article/sql-rank-functions

8) Formatting

Select format(modifiedDate, 'yyyy-MM-dd') as 'ISO formatted date', format(rate, 'c', 'en-us') as 'us currency', format(rate, 'c', 'en-gb') as 'uk currency' from employeepayhistory where year(modified date)='2004'

(or)

```
Select CONVERT(char(10),ModifiedDate,126) as 'ISO Formatted Date',
CONCAT('$' , CONVERT(varchar(12),Convert(Money,Rate))) as 'US Currency',
CONCAT('£' , CONVERT(varchar(12),Convert(Money,Rate))) as 'UK Currency'
from EmployeePayHistory
where year(ModifiedDate)  = '2004'
```

Refer : https://www.c-sharpcorner.com/blogs/format-function-in-sql-server-2012

9) Merge statement

Problem Statement:
Write a query using MERGE statement and perform delete and update operations in a single query.

Design Rules :
1. Relationship between CountryRegion and StateProvince is based on CountryRegionCode.
2. Target table for Update and Delete operation is StateProvince
3. Perform delete operation in the merge statement when CountryRegionCode is “ZZ” .
4. Perform update action in the merge statement when CountryRegionCode is “ZY” , update column Name as 'MERGE'
5. Please follow the order of operation as mentioned above. First delete, then update.

Code :

```
Merge StateProvince as target
Using(select countryregioncode, name from countryregion) as source
On target. Countryregioncode = source.countryregioncode
When matched and source.countryregioncode=’ZZ’ then delete
When matched and source.countryregioncode = ‘ZY’ then update
SET target.name=’merge’
```

10) PIVOT

```
//*SELECT *
    INTO #PayHistoryPivotResult
*// Template
```

```
FROM
 (
   SELECT
   EmployeeID, Rate, year(ModifiedDate) as ModifiedYear FROM
 EmployeePayHistory where EmployeeID <=4 ) T
PIVOT
   (SUM([Rate]) for [ModifiedYear] IN ([2004],[2002],[1997]))
//*  --TODO : Write your Pivot code here
AS PivotPayHistory;
select * from #PayHistoryPivotResult;*//
```

Refer : https://www.c-sharpcorner.com/UploadFile/f0b2ed/pivot-and-unpovit-in-sql-server

Repeating (#)

4. Using Joins

Select c.firstname + ' ' + c.lastname as name, d.name as Department , Case when e.managerid is not null then c2.firstname + ' ' + c2.lastname else 'CEO' end as manager, ad.city, sd.stateprovincecode, sd.countryregioncode from

Contact c join employee e on c.contactid = e.contactid

Join employeedepartmenthistory edh on edh.employeeid = e.employeeid

Join department d on d. departmentid = edh. departmentid

Join employeeaddress ea on ea.employeeid = e.employeeid

Join address ad on ad.addressid = ea.addressid

Leftjoin contact c2 on e.managerid = c2.contactid

Join stateprovince sd on sd.stateprovinceid = ad. Stateprovinceid

Where ad.city = 'phoenix' order by department;

2. Retrieve Rate Count Details

Select year(ratechangedate) as 'year', count(rate) as "no of rates changed", count(distinct rate) as "distinct rates" from employeepayhistory group by year(ratechangedate) having (year(datechangedate)>= '1996' and year(datechangedate)>= '1998');

8. Formatting Date

Select format (modifieddate, 'yyyy-MM-dd') as 'ISo formatted date', format (rate, 'c','en-us') as 'us currency', format(rate,'c','en-gb') as 'uk currency' from employeepayhistory where year(modifieddate)='2014';

1. **Using Joins**

Select c.lastname, c.firstname, e.employeeid, d.name as department, eph.rate from

Employee e join employeepayhistory eph on e.empid = eph.empid

Join contact c on e.contactid = c.contactid

Join employeedepartmenthistory edh on e.empid = edh.empid

Join department d on edh.deptid = d.deptid

Where c.firstname like 'P%';

5. Pagination

Select m.rownum as employeeid, m.managerid, m.title,m.gender from (select rownumber() over (order by employeeid) as rownum, managerid, title,gender from employee)m where rownum>5 and rownum>9;

7. Using Rank

Select year(modifieddate) as 'modifiedyear', rate, rank() over(partion by year(modifieddate) order by rate desc) as rateranking from employeepayhistory where year(modifieddate)>2000 order by year(modifieddate);

10. Using Pivot

```
SELECT *
    INTO #PayHistoryPivotResult
FROM
 (
   SELECT
   EmployeeID, Rate, year(ModifiedDate) as ModifiedYear FROM
```

```
 EmployeePayHistory where EmployeeID <=4 ) T
PIVOT
   (SUM([Rate]) for [ModifiedYear] IN ([2004],[2002],[1997]))
  --TODO : Write your Pivot code here
AS PivotPayHistory;
select * from #PayHistoryPivotResult;
```

3.Return employee details whose state is AB Using Procedure

```
CREATE PROCEDURE [GetEmployeeDetails]
@StateProvinceCode nvarchar(3)
AS
BEGIN
SELECT c.Lastname,c.firstname,a.city,s.StateProvinceCode
from Contact c
join Employee e on c.ContactId = e.ContactId
join EmployeeAddress ea on e.EmployeeID = ea.EmployeeID
join Address a on ea.AddressID = a.AddressID
join StateProvince s on a.StateProvinceId = s.StateProvinceId
where s.StateProvinceCode = @StateProvinceCode
END
```