--SQL Data Types

--Changing the column data types

--CAST

SELECT CAST(funding\_total\_usd AS varchar) AS funding\_total\_usd\_string

FROM tutorial.crunchbase\_companies\_clean\_date

--Subqueries

Select sub.\*

From ( Select \*

FROM tutorial.sf crime\_incidents\_2014\_01

WHERE day\_of\_week = ‘Friday’

) sub

Where sub.resolution = ‘None’

--Window functions helps to display the full table along with max\_salary with over()

select e.\*,

max(salary) over() as max\_salary

from employeedetails e

--to find out the max salary from each dept and to display all the dept using window function

--over(partition by)

select e.\*,

max(salary) over(partition by deptname) as max\_salary

from employeedetails e

--row\_number, rank, dense\_rank, lead and lag

select e.\*,

row\_number() over() as Rno

from employeedetails e;

--to assign row no acc to deptname

select e.\*,

row\_number() over(partition by deptname) as Rno

from employeedetails e;

--use case: to fetch the first 2 emp from each dept joined in the company

--first order the emp id and rank them accordingly

--row\_number

select e.\*,

row\_number() over(partition by deptname order by empid) as Rno

from employeedetails e;

--second write a sub query and where clause, the first 2 emp from each dept can be fetched

select \* from (

select e.\*,

row\_number() over(partition by deptname) as Rno

from employeedetails e) x

where x.Rno < 3;

--use case fetch the top 3 emp in each dept earning the max salary

--rank the emp in desc order using rank()

select e.\*,

rank() over(partition by deptname order by salary desc) as Rnk

from employeedetails e;

--fetch the top 3 using the subquery

select \* from (

select e.\*,

rank() over(partition by deptname order by salary desc) as Rnk

from employeedetails e) x

where x.Rnk < 4;

--dense\_rank

select e.\*,

rank() over(partition by deptname order by salary desc) as Rnk,

dense\_rank() over(partition by deptname order by salary desc) as dense\_Rnk,

row\_number() over(partition by deptname) as Rno

from employeedetails e;

--LAG windows fn

--To fetch a query to display if the salary of the employee is higher or lower or equal to the previous employee.

SELECT e.\*,

lag(salary) over (partition by deptname order by empid) AS Previous\_Salary

FROM employeedetails e;

--Passing the arguments into lag

SELECT e.\*,

lag(salary,2,0) over (partition by deptname order by empid) AS Previous\_Salary

FROM employeedetails e;

--Lead

SELECT e.\*,

lead(salary) over (partition by deptname order by empid) AS Next\_Salary

FROM employeedetails e;