use Recruitment;

select \* from ExternalCandidate;

select cCandidateCode, vFirstName, vLastName, sitestScore from ExternalCandidate;

select cCandidateCode, vFirstName+vLastName, sitestScore from ExternalCandidate;

select cCandidateCode,'Name'= vFirstName+space(1)+vLastName, sitestScore from ExternalCandidate;

select cCandidateCode,'Name'= vFirstName+space(1)+vLastName, sitestScore from ExternalCandidate;

select cCandidateCode,CandidateName= vFirstName+space(1)+vLastName, sitestScore from ExternalCandidate;

select cCandidateCode,'Candidate Name'= vFirstName+space(1)+vLastName, sitestScore from ExternalCandidate;

select cCandidateCode,vFirstName+space(1)+vLastName 'Candidate Name', sitestScore from ExternalCandidate;

select cCandidateCode, vFirstName, vLastName, sitestScore from ExternalCandidate where vFirstName='angela';

select cCandidateCode, vFirstName, vLastName, sitestScore from ExternalCandidate where cCandidateCode='000280';

select 'names:'+ vFirstName, vlastname from ExternalCandidate;

select cCandidateCode, vFirstName, vLastName, sitestScore, 'New Marks'=siTestScore+10 from ExternalCandidate;

select cCandidateCode, vFirstName, vLastName, sitestScore from ExternalCandidate where siTestScore=80;

select cCandidateCode, vFirstName, vLastName, sitestScore from ExternalCandidate where siTestScore>80 AND siTestScore<90;

select \* from ExternalCandidate where cCity='Norton' OR cCity='Mentor';

select \* from ExternalCandidate where cCity='Norton' AND siTestScore>90;

-- NOT EQUAL TO

select \* from ExternalCandidate

where ccity <> 'Norton';

-- OR Condition

select \* from ExternalCandidate

where ccity = 'Norton' or ccity = 'Mentor' or ccity = 'Dublin' or ccity = 'seattle';

-- IN Clause

select \* from ExternalCandidate

where ccity in ('Norton', 'Mentor', 'Dublin', 'seattle');

-- Comparison using >= and <=

select \* from ExternalCandidate

where sitestscore >= 80 and sitestscore <= 90;

-- Using BETWEEN

select \* from ExternalCandidate

where sitestscore between 80 and 90;

-- LIKE for pattern matching

select \* from ExternalCandidate

where vAddress like '%street%';

-- Candidates whose firstname starts with 'J'

select \* from ExternalCandidate

where vFirstname like 'J%';

-- Get all candidates who have 6 as last digit in their cCandidatecode

select \* from ExternalCandidate

where cCandidateCode like '%6';

-- Candidates whose cCandidatecode ends with 1

select \* from ExternalCandidate

where cCandidatecode like '%1';

-- Candidates whose cCandidatecode ends with 2, 6, or 8

select \* from ExternalCandidate

where cCandidatecode like '%[268]';

-- Candidates whose cCandidatecode ends with digits from 2 to 8

select \* from ExternalCandidate

where cCandidatecode like '%[2-8]';

-- Candidates whose firstname starts with 'J', 'A', or 'B' and ends with 'a', 'y', or 'h'

select \* from ExternalCandidate

where (

vFirstname like 'J%' or

vFirstname like 'A%' or

vFirstname like 'B%'

)

and (

vFirstname like '%a' or

vFirstname like '%y' or

vFirstname like '%h'

);

-- View table structure

sp\_help ExternalCandidate;

-- Create and drop index

CREATE INDEX idxstatexcan ON ExternalCandidate(cstate, ccity);

DROP INDEX idxstatexcan ON ExternalCandidate;

-- Select with ORDER BY on two columns

SELECT cstate, ccity, cCandidatecode, vFirstname, vLastname, sitestscore

FROM ExternalCandidate

ORDER BY cstate, ccity;

-- Same with descending order on second column

SELECT cstate, ccity, cCandidatecode, vFirstname, vLastname, sitestscore

FROM ExternalCandidate

ORDER BY cstate, ccity DESC;

-- Select few columns, ordered by sitestscore descending

SELECT cCandidatecode, vFirstname, sitestscore

FROM ExternalCandidate

ORDER BY sitestscore DESC;

-- Select top 10 percent candidates by score

SELECT TOP 10 PERCENT cCandidatecode, vFirstname, vLastname, sitestscore

FROM ExternalCandidate

ORDER BY sitestscore DESC;

-- Select top 5 candidates by score

SELECT TOP 5 cCandidatecode, vFirstname, sitestscore

FROM ExternalCandidate

ORDER BY sitestscore DESC;

-- Basic Aggregate Queries

select sitestscore from externalcandidate;

select

max(sitestscore),

min(sitestscore),

avg(sitestscore),

count(sitestscore)

from externalcandidate;

select

count(ccandidatecode),

count(cEmailId)

from externalcandidate;

select count(\*) from externalcandidate;

select count(ccandidatecode) from externalcandidate;

-- Group By

select

cstate,

'No of Candidates' = count(ccandidatecode)

from externalcandidate

group by cstate;

select \* from externalcandidate

order by cstate;

select

cstate,

'No of Candidates' = count(ccandidatecode),

max = max(sitestscore),

min = min(sitestscore),

avg = avg(sitestscore)

from externalcandidate

group by cstate;

-- Using WHERE and HAVING together

select

cstate,

'No of Candidates' = count(ccandidatecode),

max = max(sitestscore),

min = min(sitestscore),

avg = avg(sitestscore)

from externalcandidate

where sitestscore > 80

group by cstate

having count(ccandidatecode) >= 2;

-- Using only HAVING

select

cstate,

'No of Candidates' = count(ccandidatecode),

max = max(sitestscore),

min = min(sitestscore),

avg = avg(sitestscore)

from externalcandidate

group by cstate

having avg(sitestscore) >= 70;

--Difference between HAVING and WHERE

-- Using HAVING on GROUP BY

select

cstate,

'No of Candidates' = count(ccandidatecode),

max = max(sitestscore),

min = min(sitestscore),

avg = avg(sitestscore)

from externalcandidate

group by cstate

having cstate in ('Nevada', 'ohio', 'texas');

-- Using WHERE before GROUP BY

select

cstate,

'No of Candidates' = count(ccandidatecode),

max = max(sitestscore),

min = min(sitestscore),

avg = avg(sitestscore)

from externalcandidate

where cstate in ('Nevada', 'ohio', 'texas')

group by cstate;

-- Final Filtered Report

select

cstate,

'No of Candidates' = count(ccandidatecode),

max = max(sitestscore),

min = min(sitestscore),

avg = avg(sitestscore)

from externalcandidate

where cstate in ('Nevada', 'ohio', 'texas')

group by cstate

having count(ccandidatecode) > 2;

-- View all data from both tables

select \* from ExternalCandidate;

select \* from RecruitmentAgencies;

--Selecting candidate details with agency code

select

cCandidateCode,

vFirstname,

vLastname,

cAgencyCode

from ExternalCandidate;

-- where cAgencyCode is not null; -- Optional filter

--Selecting candidate and agency name using INNER JOIN

select

cCandidateCode,

vFirstname,

cName

from ExternalCandidate

inner join RecruitmentAgencies

on ExternalCandidate.cAgencyCode = RecruitmentAgencies.cAgencyCode;

sp\_help;

select \* from position;

SELECT cCandidateCode,vFirstName,vDescription

FROM ExternalCandidate

INNER JOIN Position

ON ExternalCandidate.cPositionCode = position.cpositionCode;

sp\_help;

select \* from ContractRecruiter;

SELECT cCandidateCode,vFirstName,cName

FROM ExternalCandidate

INNER JOIN ContractRecruiter

ON ExternalCandidate.cContractRecruiterCode = ContractRecruiter.cContractRecruiterCode;

-- Ambiguous column

select ccandidatecode, vfirstname, 'Candidate phone' = cphone, cname, 'Recruiter phone' = cphone

from ExternalCandidate

join RecruitmentAgencies

on ExternalCandidate.cAgencyCode = RecruitmentAgencies.cAgencyCode

-- Resolved (fully qualified names)

select ccandidatecode, vfirstname, 'Candidate phone' = ExternalCandidate.cphone, cname, 'Recruiter phone' = RecruitmentAgencies.cphone

from ExternalCandidate

join RecruitmentAgencies

on ExternalCandidate.cAgencyCode = RecruitmentAgencies.cAgencyCode

-- Resolved (shorter version)

select ccandidatecode, vfirstname, cname, 'Recruiter phone' = RecruitmentAgencies.cphone

from ExternalCandidate

join RecruitmentAgencies

on ExternalCandidate.cAgencyCode = RecruitmentAgencies.cAgencyCode

-- Alias

select ccandidatecode, vfirstname, cname, r.cphone

from ExternalCandidate e

join RecruitmentAgencies r

on e.cAgencyCode = r.cAgencyCode

-- Left Join

select ccandidatecode, vfirstname, cname, r.cphone

from ExternalCandidate e left outer join RecruitmentAgencies r

on e.cAgencyCode = r.cAgencyCode

-- Right Join

select ccandidatecode, vfirstname, cname, r.cphone

from ExternalCandidate e right outer join RecruitmentAgencies r

on e.cAgencyCode = r.cAgencyCode

-- Full Outer Join

select ccandidatecode, vfirstname, cname, r.cphone

from ExternalCandidate e full outer join RecruitmentAgencies r

on e.cAgencyCode = r.cAgencyCode

-- Q1: Get all candidate code and name with position name they have applied for, show all data from ExternalCandidate table

SELECT cCandidateCode,vFirstName,vDescription

FROM ExternalCandidate

LEFT JOIN Position

ON ExternalCandidate.cPositionCode = position.cpositionCode;

-- Q2: Get all candidate code and name with name of their contractor, show all data from both the tables

SELECT cCandidateCode,vFirstName,cName

FROM ExternalCandidate

FULL JOIN ContractRecruiter

ON ExternalCandidate.cContractRecruiterCode = ContractRecruiter.cContractRecruiterCode;

-- Q3: Get all candidate code and name with college name, show all data from College table

select \* from College;

select cCandidateCode, vFirstName, cCollegeName from ExternalCandidate

Right Join College on ExternalCandidate.cCollegeCode=College.cCollegeCode;

update employee

set cSupervisorCode='000001'

where cEmployeecode like '%[5-9]'

update employee

set cSupervisorCode='000002'

where cEmployeecode like '%[3-4]'

update employee

set cSupervisorCode='000001'

where cEmployeecode like '%[2]'

update employee

set cSupervisorCode='000003'

where cEmployeecode like '%1[1-2]'

--Self Join

select 'Employee'=emp.vFirstName+space(1)+emp.vLastName, 'Superior'=supr.vfirstName+space(1)+supr.vLastName

from Employee emp join employee supr

on emp.cSupervisorCode=supr.cEmployeeCode;

--cross join

select cCandidatecode , p.cpositioncode from ExternalCandidate e cross join Position p

order by cCandidateCode;

--subquery

select cCandidatecode, vFirstName , vLastName

from ExternalCandidate where cRating=(select cRating from ExternalCandidate

where vFirstName='angela')

--join substitute

select e.cCandidatecode, e.vfirstname, e.vlastname

from ExternalCandidate e

join ExternalCandidate o

on e.cRating = o.cRating

where o.vFirstName='Angela';

select cCandidatecode, vfirstname, vlastname

from ExternalCandidate

where cRating = (select cRating from ExternalCandidate where vFirstName='Angela');

select cCandidatecode, vfirstname, vlastname, cRating

from ExternalCandidate

where cRating in (select cRating from ExternalCandidate where vFirstName in ('David', 'Angela'));

--get all candidates who have got score greater than Angela and Barbara -- greater than 82 -- greater than greatest

select \* from ExternalCandidate

where sitestscore >

(select sitestscore from externalcandidate where vfirstname='Angela'

or vfirstname='Barbara');

select \* from ExternalCandidate

where sitestscore >

(select max(sitestscore) from externalcandidate where vfirstname='Angela'

or vfirstname='Barbara');

--all

select \* from ExternalCandidate

where sitestscore >

(select max(sitestscore) from externalcandidate

where vfirstname='Angela' or vfirstname='Barbara');

select \* from ExternalCandidate

where sitestscore > all (select sitestscore from externalcandidate

where vfirstname='Angela' or vfirstname='Barbara');

--any=min

select \* from ExternalCandidate

where sitestscore > any (select sitestscore from externalcandidate

where vfirstname='Angela' or vfirstname='Barbara');

--correlated subquery

--get all candidate who have scored greater than the avg sitestscore

select cCandidateCode, vFirstName, sitestscore

from externalcandidate

where sitestscore > (select avg(sitestscore) from ExternalCandidate);

--get all the candidate who have scored greater than the average score of their state

select cCandidateCode, vFirstName, sitestscore, cstate

from externalcandidate o

where sitestscore > (

select avg(sitestscore)

from ExternalCandidate i

where i.cstate = o.cstate);

--inline views

select cCandidateCode, vFirstName, sitestscore, o.cstate, Avgscore

from externalcandidate o

join (

select cstate, avg(sitestscore) 'Avgscore'

from ExternalCandidate group by cstate) i on o.cstate = i.cstate

where o.sitestscore > i.Avgscore;

-- 1. Get External Candidates if any Position has NULL cPositionCode

SELECT \*

FROM ExternalCandidate

WHERE EXISTS (

SELECT \*

FROM Position

WHERE cPositionCode IS NULL

);

-- 2. Get External Candidates if there's an Employee with vFirstName = 'Angela'

SELECT \*

FROM ExternalCandidate

WHERE EXISTS (

SELECT \*

FROM Employee

WHERE vFirstName = 'Angela'

);

-- 3. Get External Candidates if a Contract Recruiter named 'John Smith' exists

SELECT \*

FROM ExternalCandidate

WHERE EXISTS (

SELECT \*

FROM ContractRecruiter

WHERE cname = 'John Smith'

);

-- 1. Show candidates only if a Contract Recruiter named 'John Smith' exists

SELECT \*

FROM ExternalCandidate

WHERE EXISTS (

SELECT \*

FROM ContractRecruiter

WHERE cname = 'John Smith'

);

-- 2. Copy only the structure of ExternalCandidate to a new table called newexcandidate

SELECT \*

INTO newexcandidate

FROM ExternalCandidate

WHERE 1 = 0; -- Ensures only structure is copied, no data

-- 3. Create a new table newexscore with selected columns and filtered by state 'Georgia'

SELECT cCandidateCode, vFirstName, vLastName, siTestScore

INTO newexscore

FROM ExternalCandidate

WHERE cState = 'Georgia';

-- 4. Show the structure of the newexscore table

EXEC sp\_help newexscore;

-- 5. Select all data from the newexscore table

SELECT \* FROM newexscore;

-- 1. Show candidates only if a Contract Recruiter named 'John Smith' exists

SELECT \*

FROM ExternalCandidate

WHERE EXISTS (

SELECT \*

FROM ContractRecruiter

WHERE cname = 'John Smith'

);

-- 2. Copy only the structure of ExternalCandidate to a new table called newexcandidate

SELECT \*

INTO newexcandidate

FROM ExternalCandidate

WHERE 1 = 0; -- Ensures only structure is copied, no data

-- 3. Create a new table newexscore with selected columns and filtered by state 'Georgia'

SELECT cCandidateCode, vFirstName, vLastName, siTestScore

INTO newexscore

FROM ExternalCandidate

WHERE cState = 'Georgia';

-- 4. Show the structure of the newexscore table

EXEC sp\_help newexscore;

-- 5. Select all data from the newexscore table

SELECT \* FROM newexscore;

-- Valid UNION (same columns and data types)

SELECT vFirstName, vLastName, cphone

FROM externalCandidate

UNION

SELECT vFirstName, vLastName, cphone

FROM employee;

-- Valid UNION ALL (same columns and data types)

SELECT vFirstName, vLastName, cphone

FROM externalCandidate

UNION ALL

SELECT vFirstName, vLastName, cphone

FROM employee;

-- Valid INTERSECT

SELECT vFirstName, vLastName, cphone

FROM externalCandidate

INTERSECT

SELECT vFirstName, vLastName, cphone

FROM employee;

-- Valid EXCEPT

SELECT vFirstName, vLastName, cphone

FROM externalCandidate

EXCEPT

SELECT vFirstName, vLastName, cphone

FROM employee;

-- Valid UNION ALL with different column names but same structure

SELECT cname, cphone

FROM ContractRecruiter

UNION ALL

SELECT 'Name' = vFirstName + vLastName, cphone

FROM externalCandidate;

-- Just displaying all data from ContractRecruiter

SELECT \* FROM ContractRecruiter;

CREATE TABLE ExternalCandidate2 (

cCandidateCode CHAR(6) CONSTRAINT Pk\_ExternalCandidate2\_cCandidateCode PRIMARY KEY

CONSTRAINT eccd\_ck CHECK (cCandidateCode LIKE '[0-9][0-9][0-9][0-9][0-9][0-9]'),

vFirstName VARCHAR(20) NOT NULL, vLastName VARCHAR(20),

cCity CHAR(20) DEFAULT 'Mumbai' CONSTRAINT ch\_ccity

CHECK (cCity IN ('Mumbai', 'Delhi', 'Hyderabad', 'Chennai', 'Banglore')),

cPhone CHAR(15) CONSTRAINT ecp\_ck1

CHECK (cPhone LIKE '([0-9][0-9][0-9])[0-9][0-9][0-9]-[0-9][0-9][0-9][0-9]'),

cPositionCode CHAR(4) CONSTRAINT fk\_position\_cPositionCode

REFERENCES Position(cPositionCode),

dDateOfApplication DATETIME,

siPrevWorkExperience SMALLINT CONSTRAINT ch\_workexp

CHECK (siPrevWorkExperience > 0),

mPrevAnnualSalary MONEY,vEmailId VARCHAR(20) CONSTRAINT ec\_UK UNIQUE

);

DROP TABLE ExternalCandidate2;

DELETE FROM ExternalCandidate2;

sp\_help ExternalCandidate2;

INSERT INTO ExternalCandidate2

VALUES (

'000001',

'Beena',

'Shah',

DEFAULT,

'(123)123-1234',

'0001',

'06/06/2025',

1.1000000,

1000,

'beelnja@gmail.com'

);

SELECT \* FROM ExternalCandidate2;

-- Alter table to add constraint

ALTER TABLE ExternalCandidate2

ADD CONSTRAINT ec\_code\_ck000000

CHECK (cCandidateCode NOT LIKE '000000');

-- Update city name

UPDATE ExternalCandidate

SET cCity = 'Norton'

WHERE cCity = 'Norton1';

-- Set work experience to 10 for Angela

UPDATE ExternalCandidate

SET siPrevWorkExperience = 10

WHERE vFirstName = 'Angela';

-- Decrease work experience by 1 for all

UPDATE ExternalCandidate

SET siPrevWorkExperience = siPrevWorkExperience - 1;

-- Update salary of candidate using value from another record

UPDATE ExternalCandidate

SET mPrevAnnualSalary = (

SELECT mPrevAnnualSalary

FROM ExternalCandidate

WHERE vFirstName = 'Angela'

)

WHERE cCandidateCode = '000204';

-- Convert to uppercase

SELECT UPPER(vFirstName) FROM ExternalCandidate;

-- Convert to lowercase

SELECT LOWER(vFirstName) FROM ExternalCandidate;

-- Assign and use a string value

SELECT '123' AS val FROM ExternalCandidate;

-- Remove leading spaces

SELECT LTRIM(' 123') FROM ExternalCandidate;

-- Remove trailing spaces

SELECT RTRIM('123 ') FROM ExternalCandidate;

-- Length with spaces

SELECT LEN(' 123 ') FROM ExternalCandidate;

-- Length of names

SELECT LEN(vFirstName), vFirstName FROM ExternalCandidate;

-- Find position of pattern '&an'

SELECT vFirstName, PATINDEX('%an%', vFirstName)

FROM ExternalCandidate;

-- Find position of character 'A'

SELECT vFirstName, CHARINDEX('A', vFirstName)

FROM ExternalCandidate;

-- Extract 2 characters starting from position 5

SELECT vFirstName, SUBSTRING(vFirstName, 5, 2)

FROM ExternalCandidate;

-- Extract last 2 characters

SELECT vFirstName,

REVERSE(SUBSTRING(REVERSE(vFirstName), 1, 2))

FROM ExternalCandidate;

-- Extract last 2 characters using LEN

SELECT vFirstName,

SUBSTRING(vFirstName, LEN(vFirstName) - 1, 2)

FROM ExternalCandidate;