

## CSE 581 Introduction to Database Management Systems



---

Fall 2022

### Project 2 Report

#### Human Resources Database

Name: Dongze Xie

NetID: dxie03

SUID: 375402964

Date: Dec 11, 2022

---

## Abstract

This is the project report for CSE581 Project2. In this project, I designed, implemented, and tested a database called HRDB for the Recruitment branch of the HR Department. In the design section, I demonstrated my considerations and listed tables and ER diagram. In the implement section, I showed sql codes as demand. In the scenario test section, I showed my sql code for testing and it includes the necessity of the scenario in the comment. In the conclusion section, I shared my thoughts on this experience. In the appendix section, I showed screenshots during the project, and also the result for the business reports.

# Table of Contents

## Table of Contents

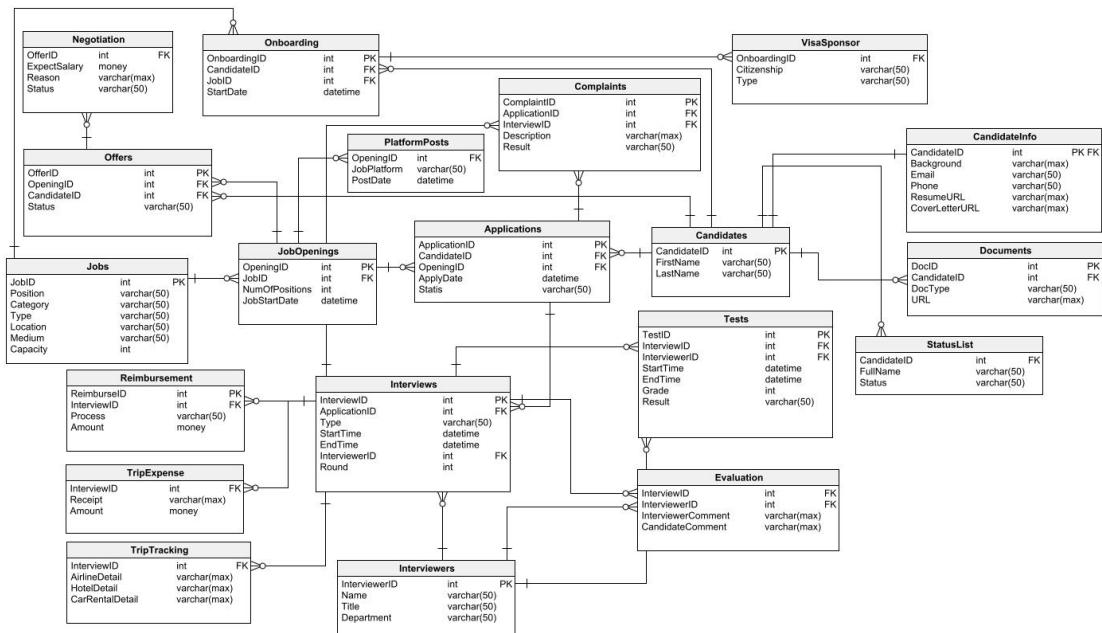
### 目录

Project 2 Report .....	1
Abstract .....	2
Table of Contents .....	3
Section A. Design .....	4
Section B. Implement .....	6
1. Create Schema, DB and tables .....	6
2. Initialize values .....	12
3. Views .....	19
4. Stored Procedures .....	21
5. User defined functions .....	23
6. Triggers .....	26
7. Transactions .....	29
8. Scripts .....	31
9. Business Reports .....	34
Section C. Scenario Test .....	37
1. Test views .....	37
2. Test User defined functions .....	39
3. Test Stored Procedures .....	42
4. Test Transactions .....	47
5. Test Trigger .....	51
6. Business Reports' test result is in Appendix .....	54
Section D. Conclusion .....	55
Appendix .....	56
1. Screenshots for database set up .....	56
2. Screenshots for create views .....	56
3. Screenshots for create user defined functions .....	58
4. Screenshots for create sp .....	60
5. Screenshots for create transactions .....	61
6. Screenshots for create triggers .....	63
7. Screenshots for create scripts .....	64
8. Screenshots for business reports .....	66

# Section A

## Section A. Design

### ER Diagram



## Descriptions

Note that if you have any doubts, you can also go to B2 to take a look at the insert value code.

**Candidates:** This is a standard table lists all candidates and their name.

**Jobs:** This is a table contains detail for each job; Position is like the title of the job;

Category is like which department it belongs to; Type can be full-time, part-time, Contract-based and summer intern; Medium can be online or onsite; Capacity is the limit of number of position for this job altogether which includes current employees.

**JobOpenings:** NumOfPositions is the number of the remaining openings, it reduces or increase automatically with the help of some triggers.

**Applications:** Status can be Pending, On progress, Rejected, Waiting(if being rejected and submit a complaint), Offer extended(approved)

**StatusList:** This is a table tracking status of candidates, blacklisted and on-call for next job opportunity included. And it can also be Pending(haven't apply yet), Applied, Round1, Round2, Round3(Final round), Rejected, Waiting, Negotiating , Onboarding.

## Section A

**CandidateInfo:** This a table contains information of each candidates.

**Interviewers:** This is a table contains information of each interviewers.

**Interviews:** One interview can be held by multiple interviewers but it is only for one candidate; it can be online or onsite; it also records the round number; the ID is special designed.

**Complaints:** The default result is handling ,when being validated correct or wrong, changed to Valid or Invalid.

**Documents:** It can show the type of content.

**Evaluation:** It records evaluation from interviewers and candidates mutually.

**Onboarding:** This table contains candidates who enter the onboard process.

**Offers:** Status can be Accepted, Declined, Negotiating, Pending(not answer and not negotiating either).

**Negotiation:** List the expect salary and reason and to be considered. Status can be Approved, Rejected and Pending, note that after reject, new negotiations on same offer can also be posted.

**PlatformPosts:** This table records the job opening hiring message is posted where and when.

**Reimbursement:** Process can be pending, verifying, proceed, rejected.

**TripExpense, TripTracking:** This two table tracks information about onsite interview trip.

**Tests:** One interview can have many tests held by same or different interviewers. Note that the ID is also special designed. The result for one interview(same time period) is same, which is depend on all tests grade within. Before test take place, grade is set to 0 and status is pending.

**VisaSponsor:** It records the citizenship and type of visa needed to be support for all foreign onboard candidates.

## Section B

### Section B. Implement

#### 1. Create Schema, DB and tables

```
USE master
GO

IF SCHEMA_ID('dxdb') IS NOT NULL
    DROP SCHEMA dxdb
GO

IF DB_ID('HRDB') IS NOT NULL
    DROP DATABASE HRDB
GO

CREATE DATABASE HRDB
GO

USE HRDB
GO

CREATE SCHEMA dxdb
GO

CREATE TABLE dxdb.Candidates(
    CandidateID int NOT NULL PRIMARY KEY,
    FirstName varchar(50) NOT NULL,
    LastName varchar(50) NOT NULL
);

CREATE TABLE dxdb.Jobs(
    JobID int NOT NULL PRIMARY KEY,
    Position varchar(50),
    Category varchar(50),
    Type varchar(50),
    Location varchar(50),
```

## Section B

```
Medium varchar(50),  
Capacity int  
);  
  
CREATE TABLE dxdb.JobOpenings(  
    OpeningID int NOT NULL PRIMARY KEY,  
    JobID int,  
    NumOfPositions int,  
    JobStartDate datetime,  
    CONSTRAINT FK_JobOpenings_Jobs FOREIGN KEY (JobID) REFERENCES dxdb.Jobs(JobID)  
);  
  
CREATE TABLE dxdb.Applications(  
    ApplicationID int NOT NULL PRIMARY KEY,  
    CandidateID int,  
    OpeningID int,  
    ApplyDate datetime,  
    Status varchar(50),  
    CONSTRAINT FK_Applications_Candidates FOREIGN KEY (CandidateID) REFERENCES  
dxdb.Candidates(CandidateID),  
    CONSTRAINT FK_Applications_JobOpenings FOREIGN KEY (OpeningID) REFERENCES  
dxdb.JobOpenings(OpeningID)  
);  
  
CREATE TABLE dxdb.StatusList(  
    CandidateID int,  
    FullName varchar(50),  
    Status varchar(50),  
    CONSTRAINT FK_StatusList_Candidates FOREIGN KEY (CandidateID) REFERENCES  
dxdb.Candidates(CandidateID)  
);  
  
CREATE TABLE dxdb.CandidateInfo(  
    CandidateID int,  
    Background varchar(max) NULL,  
    Email varchar(50),
```

## Section B

```
Phone varchar(50),  
ResumeURL varchar(max) NULL,  
CoverLetterURL varchar(max) NULL,  
CONSTRAINT FK_CandidateInfo_Candidates FOREIGN KEY (CandidateID) REFERENCES  
dxdbo.Candidates(CandidateID)  
);  
  
CREATE TABLE dxdbo.Interviewers(  
    InterviewerID int NOT NULL PRIMARY KEY,  
    Name varchar(50),  
    Title varchar(50),  
    Department varchar(50)  
);  
  
CREATE TABLE dxdbo.Interviews(  
    InterviewID int NOT NULL PRIMARY KEY,  
    ApplicationID int,  
    Type varchar(50),  
    StartTime datetime,  
    EndTime datetime,  
    InterviewerID int,  
    InterviewRound int,  
    CONSTRAINT FK_Interviews_Applications FOREIGN KEY (ApplicationID) REFERENCES  
dxdbo.Applications(ApplicationID),  
    CONSTRAINT FK_Interviews_Interviewers FOREIGN KEY (InterviewerID) REFERENCES  
dxdbo.Interviewers(InterviewerID)  
);  
  
CREATE TABLE dxdbo.Complaints(  
    ComplaintID int NOT NULL PRIMARY KEY,  
    ApplicationID int,  
    InterviewID int,  
    Description varchar(max),  
    Result varchar(50),  
    CONSTRAINT FK_Complaints_Applications FOREIGN KEY (ApplicationID) REFERENCES  
dxdbo.Applications(ApplicationID),  
    CONSTRAINT FK_Complaints_Interviews FOREIGN KEY (InterviewID) REFERENCES
```

## Section B

```
dxdb.Interviews(InterviewID)
);

CREATE TABLE dxdb.Documents(
    DocID int NOT NULL PRIMARY KEY,
    CandidateID int,
    DocType varchar(50),
    URL varchar(max),
    CONSTRAINT FK_Documents_Candidates FOREIGN KEY (CandidateID) REFERENCES
dxdb.Candidates(CandidateID)
);

CREATE TABLE dxdb.Evaluation(
    InterviewID int,
    InterviewerID int,
    InterviewerComment varchar(max),
    CandidateComment varchar(max),
    CONSTRAINT FK_Evaluation_Interviews FOREIGN KEY (InterviewID) REFERENCES
dxdb.Interviews(InterviewID),
    CONSTRAINT FK_Evaluation_Interviewers FOREIGN KEY (InterviewerID) REFERENCES
dxdb.Interviewers(InterviewerID)
);

CREATE TABLE dxdb.Onboarding(
    OnboardingID int NOT NULL PRIMARY KEY,
    CandidateID int,
    JobID int,
    StartDate datetime,
    CONSTRAINT FK_Onboarding_Candidates FOREIGN KEY (CandidateID) REFERENCES
dxdb.Candidates(CandidateID),
    CONSTRAINT FK_Onboarding_Jobs FOREIGN KEY (JobID) REFERENCES dxdb.Jobs(JobID)
);

CREATE TABLE dxdb.Offers(
    OfferID int NOT NULL PRIMARY KEY,
    OpeningID int,
    CandidateID int,
```

## Section B

```
Status varchar(50),  
    CONSTRAINT FK_Offers_JobOpenings FOREIGN KEY (OpeningID) REFERENCES  
dxdb.JobOpenings(OpeningID),  
    CONSTRAINT FK_Offers_Candidates FOREIGN KEY (CandidateID) REFERENCES  
dxdb.Candidates(CandidateID)  
);  
  
CREATE TABLE dxdb.Negotiation(  
OfferID int,  
ExpectSalary money,  
Reason varchar(max),  
Status varchar(50),  
    CONSTRAINT FK_Negotiation_Offers FOREIGN KEY (OfferID) REFERENCES  
dxdb.Offers(OfferID)  
);  
  
CREATE TABLE dxdb.PlatformPosts(  
OpeningID int,  
JobPlatform varchar(50),  
PostDate datetime,  
    CONSTRAINT FK_PlatformPosts_JobOpenings FOREIGN KEY (OpeningID) REFERENCES  
dxdb.JobOpenings(OpeningID)  
);  
  
CREATE TABLE dxdb.Reimbursement(  
ReimburseID int NOT NULL PRIMARY KEY,  
InterviewID int,  
Process varchar(50),  
Amount money,  
    CONSTRAINT FK_Reimbursement_Interviews FOREIGN KEY (InterviewID) REFERENCES  
dxdb.Interviews(InterviewID)  
);  
  
CREATE TABLE dxdb.TripExpense(  
InterviewID int,  
Receipt varchar(max),  
Amount money,
```

## Section B

```
    CONSTRAINT FK_TripExpense_Interviews FOREIGN KEY (InterviewID) REFERENCES
dxdb.Interviews(InterviewID)

);

CREATE TABLE dxdb.TripTracking(
    InterviewID int,
    AirlineDetail varchar(max),
    HotelDetail varchar(max),
    CarRentalDetail varchar(max),
    CONSTRAINT FK_TripTracking_Interviews FOREIGN KEY (InterviewID) REFERENCES
dxdb.Interviews(InterviewID)
);

CREATE TABLE dxdb.Tests(
    TestID int NOT NULL PRIMARY KEY,
    InterviewID int,
    InterviewerID int,
    StartTime datetime,
    EndTime datetime,
    Grade int,
    Result varchar(50),
    CONSTRAINT FK_Tests_Interviews FOREIGN KEY (InterviewID) REFERENCES
dxdb.Interviews(InterviewID),
    CONSTRAINT FK_Tests_Interviewers FOREIGN KEY (InterviewerID) REFERENCES
dxdb.Interviewers(InterviewerID)
);

CREATE TABLE dxdb.VisaSponsor(
    OnboardingID int,
    Citizenship varchar(50),
    Type varchar(50),
    CONSTRAINT FK_VisaSponsor_Onboarding FOREIGN KEY (OnboardingID) REFERENCES
dxdb.Onboarding(OnboardingID)
);
```

## Section B

### 2. Initialize values

USE HRDB;

```
INSERT INTO dxdb.Candidates(CandidateID,FirstName,LastName) VALUES  
(1,'Dongze','Xie'),  
(2,'Jasper','Xie'),  
(3,'Jackie','Chan'),  
(4,'Elon','Musk'),  
(5,'Taylor','Swift'),  
(6,'Ariana','Grande'),  
(7,'Ziqi','Deng'),  
(8,'Taeyeon','Kim'),  
(9,'Eren','Yeager'),  
(10,'Harry','Potter')
```

```
INSERT INTO dxdb.Jobs(JobID,Position,Category,Type,Location,Medium,Capacity) VALUES  
(101,'UI Designer','Software Development','Full-time','Syracuse','Onsite',50),  
(102,'UI Designer','Software Development','Part-time','Syracuse','Online',20),  
(103,'UI Designer','Software Development','Contract-based','Boston','Onsite',100),  
(104,'Test Engineer','Testing','Summer Intern','NYC','Onsite',30),  
(105,'Test Engineer','Testing','Summer Intern','NYC','Online',10),  
(106,'C++ Game Developer','Software Development','Full-time','San Francisco','Onsite',50),  
(107,'Android Developer','Software Development','Contract-based',null,'Online',30),  
(108,'Data Analyst','Product Marketing','Part-time','Boston','Onsite',10),  
(109,'Accountant','Finance','Full-time','Syracuse','Onsite',20),  
(110,'Backend Engineer','Software Development','Full-time','Syracuse','Online',100)
```

```
INSERT INTO dxdb.JobOpenings(OpeningID,JobID,NumOfPositions,JobStartDate) VALUES  
(201,101,20,'2023-1-1'),  
(202,101,20,'2023-8-2'),  
(203,104,10,'2023-6-1'),  
(204,105,5,'2023-6-1'),  
(205,106,30,'2023-9-1'),
```

## Section B

```
(206,107,10,'2023-12-22'),  
(207,103,50,'2023-9-1'),  
(208,108,2,'2023-12-22'),  
(209,109,10,'2023-1-1'),  
(210,110,60,'2023-7-22')
```

```
INSERT INTO dxdb.Applications(ApplicationID,CandidateID,OpeningID,ApplyDate,Status)  
VALUES  
(301,1,207,'2022-8-12','Waiting'),  
(302,1,205,'2022-8-12','On progress'),  
(303,1,210,'2022-8-12','Offer extended'),  
(304,2,203,'2022-10-22','Offer extended'),  
(305,2,202,'2022-10-22','Pending'),  
(306,10,210,'2022-9-1','Rejected'),  
(307,8,208,'2022-8-2','Offer extended'),  
(308,8,201,'2022-8-2','Pending'),  
(309,6,206,'2022-10-11','On progress'),  
(310,7,204,'2022-10-1','Pending')
```

```
INSERT INTO dxdb.StatusList(CandidateID,FullName,Status) VALUES  
(1,'Dongze Xie','Onboarding'),  
(2,'Jasper Xie','On-call for next job opportunity'),  
(3,'Jackie Chan','BlackListed'),  
(4,'Elon Musk','Onboarding'),  
(5,'Taylor Swift','BlackListed'),  
(6,'Ariana Grande','Round2'),  
(7,'Ziqi Deng','Applied'),  
(8,'Taeyeon Kim','On-call for next job opportunity'),  
(9,'Eran Yeager','Onboarding'),  
(10,'Harry Potter','Rejected')
```

## Section B

```
INSERT INTO
dxdb.CandidateInfo(CandidateID,Background,Email,Phone,ResumeURL,CoverLetterURL)VALUES
(1,'Master in CS of SU','dxie03@syr.edu','315-921-9582','url//dxie03_resume9882.pdf',null),
(2,'PhD in CS of
MIT','jasper666@gmail.com','666-666-6666','url//jasperasdfgh_666.pdf','url//jasperqwerty_
cl6.pdf'),
(3,'Former Google Web Engineer','chan333@yahoo.com','123-333-3333',null,null),
(4,'Freshman in CE
SU','elon@syr.edu','444-444-4444',null,'url//elonmusk4444ab_coverletter.docx'),
(5,null,'taylor521@gmail.com','555-555-5555',null,null),
(6,null,'ariana666@163.com','666-666-1234',null,null),
(7,'Former Twitter
Manager','gem@gmail.com','123-456-7777',null,'url//dzq7654gem_cldzq.pdf'),
(8,'Master in Statistics of Nanchang
University','taiyan@gmail.com','888-888-8888','url//ty999888abc_kty.pdf','url//ty888999abc
_mycoverletter.pdf'),
(9,null,'ailun@gmail.com','999-999-9999',null,null),
(10,'Bachelor in CS of SU','harry10@syr.edu','123-456-7890',null,null)
```

```
INSERT INTO dxdb.Interviewers(InterviewerID,Name,Title,Department) VALUES
(601,'Roger Chen','Head of Tech team','Software Development'),
(602,'Mr Lee','Java Consultant','Software Development'),
(603,'Dr Katarina','Product Manager','Software Development'),
(604,'Sam Smith','Leader of Sales team','Product Marketing'),
(605,'Dongze Real','CEO','Human Resource'),
(606,'Miss Sara','Leader of HR team','Human Resource'),
(607,'Andrew Sheeran',null,'Human Resource'),
(608,'John Snow',null,'Design and Art'),
(609,'Mrs Elena','Senior Software Engineer','Software Development'),
(610,'Dr Xie','Head of Finance Department','Finance')
```

```
INSERT INTO
dxdb.Interviews(InterviewID,ApplicationID,Type,StartTime,EndTime,InterviewerID,InterviewR
ound) VALUES
(401,307,'Online','2022-11-11 13:00:00','2022-11-11 16:00',607,1),
```

## Section B

```
(7402,307,'Online','2022-11-20 13:00:00','2022-11-20 19:00',607,2),  
(1402,307,'Online','2022-11-20 13:00:00','2022-11-20 19:00',601,2),  
(403,307,'Onsite','2022-12-1 15:00:00','2022-12-1 17:00:00',606,3), /*approved*/  
(404,304,'Online','2022-11-11 13:00:00','2022-11-11 15:00',601,3), /*approved*/  
(9405,303,'Online','2022-11-20 13:00:00','2022-11-20 19:00',609,2),  
(2405,303,'Online','2022-11-20 13:00:00','2022-11-20 19:00',602,2),  
(406,303,'Onsite','2022-12-2 15:00:00','2022-12-2 19:00:00',605,3), /*approved*/  
(407,301,'Online','2022-11-11 13:00:00','2022-11-11 16:00',608,1), /*waiting*/  
(408,302,'Onsite','2022-12-20 13:00:00','2022-12-20 19:00',601,2), /*on progress*/  
(409,306,'Online','2022-11-1 13:00:00','2022-11-1 16:00:00',609,1), /*reject*/  
(9410,309,'Online','2022-11-5 13:00:00','2022-11-5 19:00:00',609,2), /*on progress*/  
(1410,309,'Online','2022-11-5 13:00:00','2022-11-5 19:00:00',601,2) /*on progress*/
```

```
INSERT INTO dxdb.Complaints(ComplaintID,ApplicationID,InterviewID,Description,Result)  
VALUES
```

(1,301,407,

'The interviewer focus on design skill rather frontend programming ability. It is totally unfair and makes no sense!',

'Handling'),

(2,306,409,

'I think the questions are not common and even meaningless.

Furthermore the interviewer has accent, so that we cannot communicate well.',

'Invalid'),

(3,307,1402,

'I had network issue during interview. That had been recorded, and I deserve another chance!',

'Valid') /\*status changed, new time updated\*/

```
INSERT INTO dxdb.Documents(DocID,CandidateID,DocType,URL) VALUES
```

(501,1,'Resume','url//dxie03\_resume9882.pdf'),

(502,2,'Resume','url//jasperasdfgh\_666.pdf'),

(503,2,'Cover Letter','url//jasperqwerty\_cl6.pdf'),

(504,4,'Cover Letter','url//elonmusk4444ab\_coverletter.docx'),

(505,7,'Cover Letter','url//dzq7654gem\_cldzq.pdf'),

(506,8,'Resume','url//ty999888abc\_kty.pdf'),

## Section B

```
(507,8,'Cover Letter','url//ty888999abc_mycoverletter.pdf'),  
(508,1,'Disability Doc with Signature','url//dxie03_selfidentify789.pdf'),  
(509,1,'Required Files','url//dxie03aaa_package.zip'),  
(510,2,'Disability Doc with Signature','url//jasperqwer_nodisab777.pdf'),  
(511,2,'Required Files','url//jasperxyz6abc_id.zip'),  
(512,8,'Disability Doc with Signature','url//taeyeon1989_sign.pdf'),  
(513,8,'Required Files','url//ty898989aaa_allpaperworks.zip')
```

```
INSERT INTO  
dxdb.Evaluation(InterviewID,InterviewerID,InterviewerComment,CandidateComment)  
VALUES  
  
(401,607,'Qualified','Nice experience!'),  
(7402,607,'Qualified','Nice experience!'),  
(1402,601,'Excellent','Nice experience!'),  
(403,606,'This candidate is outstanding!', 'Interviewer is very kind!'),  
(404,601,'This candidate matches our need!', 'Overall good'),  
(9405,609,'Nice communication skill and superb algorithm ability!', 'I like this interviewer!'),  
(2405,602,'Amazing response for every question and can perfectly complete our test!', 'I suggest we shuold take a break during the second and third part of interview'),  
(406,605,'Talented, polite and has potential!', 'Best interview experience ever!'),  
(407,608,'The candidate and I cannot share the same sight of Arts. ', 'This interviewer was too subjective and got the focus of the test wrong!'),  
(408,601,null,null),  
(409,609,'Poor performance on answering my question.', 'The interviewer has strong accent, we can barely communicate!'),  
(9410,609,null,null),  
(1410,601,null,null)
```

```
INSERT INTO dxdb.Onboarding(OnboardingID,CandidateID,JobID,StartDate) VALUES  
(1,1,110,'2023-7-22'),  
(2,4,105,'2023-6-1'),  
(3,9,106,'2023-9-1')
```

```
INSERT INTO dxdb.Offers(OfferID,OpeningID,CandidateID,Status) VALUES  
(1,210,1,'Accepted'),
```

## Section B

```
(2,208,8,'Declined'),  
(3,203,2,'Negotiating')
```

```
INSERT INTO dxdb.Negotiation(OfferID,ExpectSalary,Reason,Status) VALUES  
(1,1200000.00,'Rejected'),  
(1,98000.00,'Reasonable paid for New Grad','Approved'),  
(2,100000.00,'Average condition in this field','Rejected'),  
(3,105000.00,'That will cover my shuttle expense and apartment rent.','Pending')
```

```
INSERT INTO dxdb.PlatformPosts(OpeningID,JobPlatform,PostDate) VALUES  
(201,'LinkedIn','2022-6-1'),  
(201,'Handshake','2022-6-1'),  
(201,'Company Mainpage','2022-6-1'),  
(201,'Indeed','2022-6-18'),  
(202,'LinkedIn','2022-7-1'),  
(202,'Handshake','2022-7-1'),  
(202,'Company Mainpage','2022-7-1'),  
(202,'Indeed','2022-7-1'),  
(203,'LinkedIn','2022-6-20'),  
(204,'LinkedIn','2022-6-20'),  
(205,'Company Mainpage','2022-5-21'),  
(206,'LinkedIn','2022-6-1'),  
(206,'Handshake','2022-6-1'),  
(207,'Simplified Jobs','2022-6-18'),  
(207,'Company Mainpage','2022-6-18'),  
(208,'LinkedIn','2022-7-1'),  
(209,'Company Mainpage','2022-6-1'),  
(210,'LinkedIn','2022-6-10'),  
(210,'Handshake','2022-6-10'),  
(210,'Company Mainpage','2022-6-10'),  
(210,'Indeed','2022-6-10')
```

```
INSERT INTO dxdb.Reimbursement(ReimburseID,InterviewID,Process,Amount) VALUES  
(1,403,'Verifying',0.00),
```

## Section B

```
(2,406,'Proceed',600.00),  
(3,408,'Pending',0.00)
```

```
INSERT INTO dxdb.TripExpense(InterviewID,Receipt,Amount) VALUES  
(403,'url//receipt#aptx4869.pdf',850.00),  
(406,'url//receipt#qwer080808.pdf',500.00),  
(408,null,0.00)
```

```
INSERT INTO dxdb.TripTracking(InterviewID,AirlineDetail,HotelDetail,CarRentalDetail) VALUES  
(403,'https://www.jetblue.com/airlineinfo#aptx4869','https://www.airbnb.com/rooms#qwer  
666',  
'https://www.hertz.com/rentacar/info#123456',  
(406,'https://www.expedia.com/flight/19981234','https://www.booking.com/order#asd28',  
'https://www.kayak.com/ertfghvbn6789',  
(408,null,null,null)
```

```
INSERT INTO dxdb.Tests(TestID,InterviewID,InterviewerID,StartTime,EndTime,Grade,Result)  
VALUES  
(4011,401,607,'2022-11-11 13:00:00','2022-11-11 15:30:00',100,'Pass'),  
(4021,7402,607,'2022-11-20 13:00:00','2022-11-20 15:00:00',98,'Pass'),  
(4022,1402,601,'2022-11-20 15:30:00','2022-11-20 17:00:00',90,'Pass'),  
(4023,1402,601,'2022-11-20 17:00:00','2022-11-20 18:30:00',92,'Pass'),  
(4051,9405,609,'2022-11-20 13:00:00','2022-11-20 15:00:00',99,'Pass'),  
(4052,9405,609,'2022-11-20 15:00:00','2022-11-20 17:00:00',99,'Pass'),  
(4053,2405,602,'2022-11-20 17:00:00','2022-11-20 19:00:00',99,'Pass'),  
(4071,407,608,'2022-11-11 13:30:00','2022-11-11 16:00:00',65,'Fail'),  
(4081,408,601,'2022-12-20 13:30:00','2022-12-20 18:30:00',0,'Pending'),  
(4091,409,609,'2022-11-1 13:00:00','2022-11-1 15:30:00',50,'Fail'),  
(4101,9410,609,'2022-11-5 13:00:00','2022-11-5 15:00:00',0,'Pending'),  
(4102,1410,601,'2022-11-5 15:00:00','2022-11-5 17:00:00',0,'Pending'),  
(4103,1410,601,'2022-11-5 17:00:00','2022-11-5 18:30:00',0,'Pending')
```

```
INSERT INTO dxdb.VisaSponsor(OnboardingID,Citizenship,Type) VALUES  
(1,'China','H-1B'),  
(3,'Japan','F1-OPT')
```

## Section B

### 3. Views

#### 3.1 view ‘FindOpeningsInSyr’

```
/*view1: find all job openings located in Syracuse*/
USE HRDB;
GO

IF OBJECT_ID('FindOpeningsInSyr') IS NOT NULL
DROP VIEW FindOpeningsInSyr
GO

CREATE VIEW FindOpeningsInSyr AS
    SELECT OpeningID, Position, Type, Medium, JobStartDate, NumOfPositions AS [Remain Openings]
    FROM dxdb.Jobs JOIN dxdb.JobOpenings
    ON dxdb.Jobs.JobID=dxdb.JobOpenings.JobID
    WHERE Location='Syracuse'
GO
```

#### 3.2 view ‘CheckBgInFinalRound’

```
/*view2: Check candidates' background who had enter the final(3rd) interview*/
USE HRDB;
GO

IF OBJECT_ID('CheckBgInFinalRound') IS NOT NULL
DROP VIEW CheckBgInFinalRound
GO

CREATE VIEW CheckBgInFinalRound AS
    SELECT FirstName+' '+LastName AS [full name], Background
    FROM dxdb.Candidates JOIN dxdb.CandidateInfo
        ON dxdb.Candidates.CandidateID = dxdb.CandidateInfo.CandidateID
    JOIN dxdb.Applications ON dxdb.Applications.CandidateID=
dxdb.Candidates.CandidateID
    JOIN dxdb.Interviews ON dxdb.Interviews.ApplicationID =
```

## Section B

```
dxdb.Applications.ApplicationID  
WHERE InterviewRound = 3;  
GO
```

### 3.3 view ‘CollectOnboardDoc’

```
USE HRDB;  
GO  
  
IF OBJECT_ID('CollectOnboardDoc') IS NOT NULL  
DROP VIEW CollectOnboardDoc  
GO  
  
CREATE VIEW CollectOnboardDoc AS  
URL  
    SELECT FirstName+' '+LastName AS [candidate full name], DocType AS Type,  
    FROM dxdb.Onboarding JOIN dxdb.Documents  
    ON dxdb.Onboarding.CandidateID = dxdb.Documents.CandidateID  
    JOIN dxdb.Candidates ON dxdb.Candidates.CandidateID =  
dxdb.Onboarding.CandidateID  
    UNION  
    SELECT FirstName+' '+LastName AS [candidate full name], 'Email' AS Type,  
Email AS URL  
    FROM dxdb.CandidateInfo JOIN dxdb.Onboarding  
    ON dxdb.CandidateInfo.CandidateID = dxdb.Onboarding.CandidateID  
    JOIN dxdb.Candidates ON dxdb.Candidates.CandidateID =  
dxdb.CandidateInfo.CandidateID  
GO
```

### 3.4 view ‘ListTestTimeDuration’

```
/*view4: list all tests with their time duration which was in an completed online  
interview */  
  
USE HRDB;  
GO  
  
IF OBJECT_ID('ListTestTimeDuration') IS NOT NULL  
DROP VIEW ListTestTimeDuration
```

## Section B

GO

```
CREATE VIEW ListTestTimeDuration AS
    SELECT TestID,
        CAST(DATEDIFF(MINUTE,dxdb.Tests.StartTime,dxdb.Tests.EndTime) AS
VARCHAR)+' minutes' AS [time duration],Grade
    FROM dxdb.Tests JOIN dxdb.Interviews
    ON dxdb.Tests.InterviewID = dxdb.Interviews.InterviewID
    WHERE Type='Online' AND Result != 'Pending'
```

GO

## 4. Stored Procedures

### 4.1 spApplyTotal

```
/*sp1: check how many candidates applied a specific jobOpening*/
USE HRDB;
GO

IF OBJECT_ID('dxdb.spApplyTotal')IS NOT NULL
DROP PROC dxdb.spApplyTotal
GO

CREATE PROC dxdb.spApplyTotal
    @OpeningID int,
    @ApplyTotal int OUTPUT
AS
    SELECT @ApplyTotal = COUNT(*)
    FROM dxdb.Applications
    WHERE OpeningID = @OpeningID
GO
```

### 4.2 spAddInterviewer

```
/*sp2: add a new interviewer and fill in his/her info*/
USE HRDB;
```

## Section B

GO

```
IF OBJECT_ID('dxdb.spAddInterviewer') IS NOT NULL
DROP PROC dxdb.spAddInterviewer
GO

CREATE PROC dxdb.spAddInterviewer
    @InterviewerID int,
    @Name varchar(50),
    @Title varchar(50),
    @Department varchar(50)
AS
IF EXISTS(SELECT * FROM dxdb.Interviewers WHERE InterviewerID = @InterviewerID)
    THROW 50001, 'InterviewerID already exists!',1;
ELSE
    INSERT INTO dxdb.Interviewers
    VALUES (@InterviewerID, @Name, @Title, @Department);
```

### 4.3 spUpdateGrade

```
/*sp3 update the grade of a specific test*/
USE HRDB;
GO

IF OBJECT_ID('dxdb.spUpdateGrade') IS NOT NULL
DROP PROC dxdb.spUpdateGrade
GO

CREATE PROC dxdb.spUpdateGrade
    @TestID int , @Grade int
AS
    IF @TestID IS NULL OR (SELECT COUNT(*) FROM dxdb.Tests WHERE TestID = @TestID) = 0
        THROW 50002, 'Not such TestID exists!',1;
    ELSE
```

## Section B

```
UPDATE dxdb.Tests SET Grade = @Grade WHERE TestID = @TestID  
GO
```

### 4.4 spTrackOnsiteInterview

```
/*sp4: track a specific onsite interview*/  
  
USE HRDB;  
GO  
  
  
IF OBJECT_ID('dxdb.spTrackOnsiteInterview') IS NOT NULL  
DROP PROC dxdb.spTrackOnsiteInterview  
GO  
  
  
CREATE PROC dxdb.spTrackOnsiteInterview  
    @InterviewID int  
AS  
    IF (SELECT Type FROM dxdb.Interviews WHERE InterviewID = @InterviewID)='Online'  
        THROW 50003, 'This is an online interview!',1;  
    ELSE  
        SELECT dxdb.TripTracking.InterviewID, AirlineDetail AS [Airline  
reservation detail],  
               HotelDetail AS [Hotel reservation detail], CarRentalDetail AS [Car  
rental detail],  
               Receipt, Amount AS [Expense Total]  
        FROM dxdb.TripTracking JOIN dxdb.TripExpense  
        ON dxdb.TripTracking.InterviewID = dxdb.TripExpense.InterviewID  
        WHERE dxdb.TripTracking.InterviewID = @InterviewID  
GO
```

## 5. User defined functions

### 5.1 fnGetResume

```
/*fn1: get the resume url of certain candidate*/  
  
USE HRDB;  
GO
```

## Section B

```
IF OBJECT_ID('dxdb.fnGetResume') IS NOT NULL
DROP FUNCTION dxdb.fnGetResume
GO

CREATE FUNCTION dxdb.fnGetResume
    (@CandidateID int)
    RETURNS VARCHAR(50)
BEGIN
    RETURN(
        SELECT ISNULL(ResumeURL, 'Not uploaded yet!') FROM dxdb.CandidateInfo
        WHERE CandidateID = @CandidateID
    );
END;
GO
```

### 5.2 fnGetInterviewerFeedBack

```
/*fn2: get the evalution from a candidate towards a certain interviewer,
and also the complaints on an interview which the interviewer participated*/
USE HRDB;
GO
```

```
IF OBJECT_ID('dxdb.fnGetInterviewerFeedBack') IS NOT NULL
DROP FUNCTION dxdb.fnGetInterviewerFeedBack
GO

CREATE FUNCTION dxdb.fnGetInterviewerFeedBack
    (@InterviewerID int)
    RETURNS table
    RETURN(
        SELECT Name AS [Interviewer Name], CONVERT(date,StartTime) AS [date of
interview], CandidateComment
        FROM dxdb.Interviewers JOIN dxdb.Evaluation
        ON dxdb.Interviewers.InterviewerID = dxdb.Evaluation.InterviewerID
        JOIN dxdb.Interviews ON dxdb.Interviews.InterviewerID =
```

## Section B

```
dxdb.Interviewers.InterviewerID  
    WHERE dxdb.Interviewers.InterviewerID = @InterviewerID  
    UNION  
        SELECT Name AS [Interviewer Name], CONVERT(date,StartTime) AS [date of  
interview], Description  
        FROM dxdb.Interviewers JOIN dxdb.Interviews  
            ON dxdb.Interviews.InterviewerID = dxdb.Interviewers.InterviewerID  
            JOIN dxdb.Complaints ON dxdb.Complaints.InterviewID =  
dxdb.Interviews.InterviewID  
    WHERE dxdb.Interviewers.InterviewerID = @InterviewerID  
);  
GO
```

### 5.3 fnListPostAfter

```
/*fn3: List all the job opening posts after certain date*/  
USE HRDB;  
GO  
  
IF OBJECT_ID('dxdb.fnListPostAfter') IS NOT NULL  
DROP FUNCTION dxdb.fnListPostAfter  
GO  
  
CREATE FUNCTION dxdb.fnListPostAfter  
    (@afterdate datetime)  
    RETURNS table  
  
    RETURN(  
        SELECT * FROM dxdb.PlatformPosts  
        WHERE PostDate > @afterdate  
    );  
GO
```

## Section B

### 5.4 fnGetMostRemain

```
/*fn4: get the current job opening with the most remaining positions*/
USE HRDB;
GO

IF OBJECT_ID('dxdb.fnGetMostRemain') IS NOT NULL
DROP FUNCTION dxdb.fnGetMostRemain
GO

CREATE FUNCTION dxdb.fnGetMostRemain()
RETURNS int

BEGIN
    RETURN(SELECT OpeningID FROM dxdb.JobOpenings
        WHERE NumOfPositions =
            (SELECT MAX(NumOfPositions) FROM dxdb.JobOpenings));
END;
GO
```

## 6. Triggers

### 6.1 trigReduceOpening

```
/*trig1: every time a position is filled by a candidate,
the number of job openings for that position should be reduced by 1*/
USE HRDB;
GO

DROP TRIGGER IF EXISTS dxdb.trigReduceOpening;
GO

CREATE TRIGGER dxdb.trigReduceOpening
ON dxdb.Applications
AFTER UPDATE
AS
```

## Section B

```
IF (SELECT Status FROM inserted) = 'Offer extended'
BEGIN
    UPDATE dxdb.JobOpenings
    SET NumOfPositions = NumOfPositions - 1
    WHERE dxdb.JobOpenings.OpeningID =
        (SELECT inserted.OpeningID
         FROM dxdb.JobOpenings JOIN inserted ON dxdb.JobOpenings.OpeningID
         = inserted.OpeningID)
END;
```

### 6.2 trigRestoreNumOfPosition

```
/*trig2: every time an offer is declined,
the correspoding job opening should increased by 1*/
USE HRDB;
GO

DROP TRIGGER IF EXISTS dxdb.trigRestoreNumOfPosition;
GO

CREATE TRIGGER dxdb.trigRestoreNumOfPosition
ON dxdb.Offers
AFTER UPDATE
AS
IF (SELECT Status FROM inserted) = 'Declined'
BEGIN
    UPDATE dxdb.JobOpenings
    SET NumOfPositions = NumOfPositions + 1
    WHERE dxdb.JobOpenings.OpeningID =
        (SELECT inserted.OpeningID
         FROM dxdb.JobOpenings JOIN inserted ON dxdb.JobOpenings.OpeningID
         = inserted.OpeningID)
END;
```

## Section B

### 6.3 trigReApply

```
/*trig3: after a rejected candidate re-apply, start a brand new status flow*/  
USE HRDB;  
GO  
  
DROP TRIGGER IF EXISTS dxdb.trigReApply;  
GO  
  
CREATE TRIGGER dxdb.trigReApply  
ON dxdb.Applications  
AFTER INSERT  
AS  
IF (SELECT Status FROM dxdb.Applications WHERE CandidateID =  
    (SELECT CandidateID FROM inserted)  
    AND OpeningID = (SELECT OpeningID FROM inserted)  
    AND ApplicationID != (SELECT ApplicationID FROM inserted)) = 'Rejected'  
    /*Only take place when re-apply the same opening which has been rejected  
in the past*/  
  
BEGIN  
    UPDATE dxdb.StatusList SET Status='Applied'  
    WHERE CandidateID = (SELECT CandidateID FROM inserted)  
END;
```

### 6.4 trigInvalidComplaint

```
/*trig4: After the Complaints department finds that the  
complaint is invalid, status is changed to rejected */  
USE HRDB;  
GO  
  
DROP TRIGGER IF EXISTS dxdb.trigInvalidComplaint;  
GO  
  
CREATE TRIGGER dxdb.trigInvalidComplaint
```

## Section B

```
ON dxdb.Complaints
AFTER UPDATE
AS
IF (SELECT Result FROM inserted)='Invalid'
BEGIN
    UPDATE dxdb.Applications SET Status = 'Rejected'
    WHERE dxdb.Applications.ApplicationID =
        (SELECT ApplicationID FROM inserted)
END;
```

## 7. Transactions

### 7.1 tran add a new candidate

```
/*tran1: add a new candidate*/
USE HRDB;
GO

BEGIN TRAN
    INSERT INTO dxdb.Candidates (CandidateID,FirstName,LastName) VALUES
        (11,'Charlie','Puth');

    INSERT INTO
        dxdb.CandidateInfo(CandidateID,Background,Email,Phone,ResumeURL,CoverLetterURL)
    VALUES
        (11,'Junior FrontEnd engineer in
Amazon','cp11@gmail.com','111-4567-1234',null,'url//charlie11qwer_cqcl.pdf');

    INSERT INTO dxdb.Documents(DocID,CandidateID,DocType,URL) VALUES
        (514,11,'Cover Letter','url//charlie11qwer_cqcl.pdf')

COMMIT TRAN
```

### 7.2 tran delete an inactive candidate

```
/*tran2: delete an inactive candidate(no application, without resume, not on the
special list)*/
USE HRDB;
GO
```

## Section B

```
BEGIN TRAN;

    DELETE FROM dxdb.Documents WHERE CandidateID = 11
    DELETE FROM dxdb.CandidateInfo WHERE CandidateID = 11
    DELETE FROM dxdb.Candidates WHERE CandidateID = 11

    IF @@ROWCOUNT > 3
        BEGIN
            ROLLBACK TRAN;
            PRINT 'This affects more than expected, deletions rolled back!';
        END
    ELSE
        BEGIN
            COMMIT TRAN;
            PRINT 'Deletions committed to the database!';
        END;
```

### 7.3 tran delete an interviewer

```
/*tran3: delete an interviewer who has never participated any interview*/
USE HRDB;
GO

BEGIN TRAN;

IF (SELECT COUNT(*) FROM dxdb.Interviews WHERE InterviewerID = 603)=0
    AND (SELECT COUNT(*) FROM dxdb.Evaluation WHERE InterviewerID = 603)=0
    AND (SELECT COUNT(*) FROM dxdb.Tests WHERE InterviewerID = 603)=0
BEGIN
    DELETE FROM dxdb.Interviewers WHERE InterviewerID = 603;
    IF @@ROWCOUNT >1
        BEGIN
            ROLLBACK TRAN;
            PRINT 'This affects more than one row! Deletion rolled back!';
        END;
    ELSE
```

## Section B

```
BEGIN  
    COMMIT TRAN;  
    PRINT 'Deletion success!';  
    END;  
  
ELSE  
    COMMIT TRAN;  
  
GO
```

### 7.4 tran process an application

```
/*tran4: process an application*/  
USE HRDB;  
GO  
  
BEGIN TRAN  
    UPDATE dxdb.Applications SET Status='On progress' WHERE ApplicationID = 310;  
  
    INSERT INTO dxdb.Interviews  
(InterviewID,ApplicationID,Type,StartTime,EndTime,InterviewerID,InterviewRound)  
VALUES  
(411,310,'Online','2022-11-23 13:00:00','2022-11-23 16:00:00',606,1);  
  
    INSERT INTO dxdb.Tests  
(TestID,InterviewID,InterviewerID,StartTime,EndTime,Grade,Result) VALUES  
(4111,411,606,'2022-11-23 13:00:00','2022-11-23 15:40:00',0,'Pending');  
  
COMMIT TRAN
```

## 8. Scripts

### 8.1 create roles

```
/*create roles*/
```

## Section B

```
/*role1: InterviewAdmin*/  
DROP ROLE InterviewAdmin;  
  
CREATE ROLE InterviewAdmin;  
GRANT UPDATE, INSERT ON dxdb.Interviewers TO InterviewAdmin;  
GRANT UPDATE, INSERT ON dxdb.Interviews TO InterviewAdmin;  
GRANT UPDATE, INSERT ON dxdb.Tests TO InterviewAdmin;  
GRANT UPDATE, INSERT ON dxdb.Evaluation TO InterviewAdmin;  
  
ALTER ROLE db_datareader ADD MEMBER InterviewAdmin  
  
/*role2: DocAdmin*/  
DROP ROLE DocAdmin;  
  
CREATE ROLE DocAdmin;  
GRANT UPDATE, INSERT ON dxdb.Documents TO DocAdmin;  
  
ALTER ROLE db_datareader ADD MEMBER DocAdmin  
  
/*role3: OnboardingTeam*/  
DROP ROLE OnboardTeam;  
  
CREATE ROLE OnboardTeam;  
GRANT UPDATE, INSERT ON dxdb.Onboarding TO OnboardTeam;  
GRANT UPDATE, INSERT ON dxdb.Offers TO OnboardTeam;  
GRANT UPDATE, INSERT ON dxdb.Negotiation TO OnboardTeam;  
GRANT UPDATE, INSERT ON dxdb.StatusList TO OnboardTeam;  
GRANT UPDATE, INSERT ON dxdb.VisaSponsor TO OnboardTeam;  
  
ALTER ROLE db_datareader ADD MEMBER OnboardTeam  
  
/*role4: ComplaintTeam*/  
DROP ROLE ComplaintTeam;  
  
CREATE ROLE ComplaintTeam;
```

## Section B

```
GRANT UPDATE, INSERT ON dxdb.Complaints TO ComplaintTeam;
GRANT UPDATE, INSERT ON dxdb.Applications TO ComplaintTeam;
GRANT UPDATE, INSERT ON dxdb.Interviews TO ComplaintTeam;
GRANT UPDATE, INSERT ON dxdb.StatusList TO ComplaintTeam;
GRANT UPDATE, INSERT ON dxdb.Tests TO ComplaintTeam;

ALTER ROLE db_datareader ADD MEMBER ComplaintTeam
```

### 8.2 script1

```
/*script1: create Login AdminInterviewer, user DrXie */
CREATE LOGIN AdminInterviewer WITH PASSWORD = '101010',
    DEFAULT_DATABASE = HRDB;
GO

CREATE USER DrXie FOR LOGIN AdminInterviewer;
ALTER ROLE InterviewAdmin ADD MEMBER DrXie
ALTER ROLE DocAdmin ADD MEMBER DrXie
```

### 8.3 script2

```
/*script2: create Login AdminOnboarding, user Andrew */
CREATE LOGIN AdminOnboarding WITH PASSWORD = '8000',
    DEFAULT_DATABASE = HRDB;
GO

CREATE USER Andrew FOR LOGIN AdminOnboarding;
ALTER ROLE OnboardTeam ADD MEMBER Andrew
ALTER ROLE DocAdmin ADD MEMBER Andrew
```

### 8.4 script3

```
/*script3: create Login AdminValidate, user IU */
CREATE LOGIN AdminValidate WITH PASSWORD = '987654',
    DEFAULT_DATABASE = HRDB;
GO
```

## Section B

```
CREATE USER IU FOR LOGIN AdminValidate;
ALTER ROLE ComplaintTeam ADD MEMBER IU
```

### 8.5 Script4

```
/*script4: create Login DomainAdmin2, user dxie03*/
CREATE LOGIN DomainAdmin2 WITH PASSWORD = '030303',
    DEFAULT_DATABASE = HRDB;
GO

CREATE USER dxie03 FOR LOGIN DomainAdmin2;
ALTER ROLE ComplaintTeam ADD MEMBER dxie03
ALTER ROLE OnboardTeam ADD MEMBER dxie03
ALTER ROLE DocAdmin ADD MEMBER dxie03
ALTER ROLE InterviewAdmin ADD MEMBER dxie03
```

## 9. Business Reports

### 9.1 report1

```
/*business report1: find out which type of medium of job is more popular*/
USE HRDB;
GO

SELECT COUNT(*) AS [#applications for online job] FROM dxdb.Applications
WHERE dxdb.Applications.OpeningID IN
    (SELECT OpeningID FROM dxdb.JobOpenings JOIN dxdb.Jobs
        ON dxdb.JobOpenings.JobID = dxdb.Jobs.JobID
        WHERE Medium='Online')
GO

SELECT COUNT(*) AS [#applications for onsite job] FROM dxdb.Applications
WHERE dxdb.Applications.OpeningID IN
```

## Section B

```
(SELECT OpeningID FROM dxdb.JobOpenings JOIN dxdb.Jobs  
    ON dxdb.JobOpenings.JobID = dxdb.Jobs.JobID  
    WHERE Medium='Onsite')  
GO
```

### 9.2 report2

```
/*business report2: list all the average score who passed the whole interview.  
Note that: a interview may contain many tests with same or different interviewer,  
but by ID we can deduct which of them belongs to which whole interview(share the  
same time period)*/
```

```
USE HRDB;  
GO
```

```
SELECT (TestID/10) AS InterviewBaseNum , AVG(Grade) AS [average score]  
    FROM dxdb.Tests  
    WHERE Result = 'pass'  
    GROUP BY (TestID/10)  
    ORDER BY (TestID/10) ASC
```

### 9.3 report3

```
/*business report3: find out is there a relationship  
between candidates' response time and post platform  
*/  
USE HRDB;  
GO
```

```
SELECT JobPlatform, AVG(DATEDIFF(day,PostDate, ApplyDate)) AS [AVGResponseTime  
(days)]  
    FROM dxdb.PlatformPosts JOIN dxdb.Applications
```

## Section B

```
ON dxdb.PlatformPosts.OpeningID = dxdb.Applications.OpeningID  
GROUP BY JobPlatform  
ORDER BY AVG(DATEDIFF(day, PostDate, ApplyDate)) ASC
```

### 9.4 report4

```
/*business report: find out which job type has most current employees  
and how many current employees in total do we have*/  
USE HRDB;  
GO  
  
SELECT Type, SUM(Capacity - NumOfPositions) AS [current num of employees]  
FROM dxdb.Jobs JOIN dxdb.JobOpenings  
    ON dxdb.Jobs.JobID = dxdb.JobOpenings.JobID  
GROUP BY Type WITH ROLLUP  
ORDER BY Type DESC
```

### Section C. Scenario Test

#### 1. Test views

##### 1.1 Test code

testview1-----

```
USE HRDB;
```

```
GO
```

```
/*select all to show the result of the view*/
```

```
SELECT * FROM FindOpeningsInSyr;
```

```
GO
```

testview2-----

```
USE HRDB;
```

```
GO
```

```
/*select all from the view to check the result*/
```

```
SELECT * FROM CheckBgInFinalRound;
```

```
GO
```

testview3-----

```
USE HRDB;
```

```
GO
```

```
/*select all to show the result of the view*/
```

```
SELECT * FROM CollectOnboardDoc;
```

testview4-----

```
USE HRDB;
```

```
GO
```

```
/*select all to show the result of the view*/
```

```
SELECT * FROM ListTestTimeDuration;
```

## Section C

### 1.2 Test Results

The screenshot displays two Microsoft SQL Server Management Studio (SSMS) windows side-by-side, both connected to the HRDB database.

**Top Window:**

- Query:** `/*select all to show the result of the view*/  
SELECT * FROM FindOpeningsInSyr;  
GO`
- Results:** A table titled "Results" showing four rows of data:

OpeningID	Position	Type	Medium	Start Date	Remain Openings
201	UI Designer	Full-time	Onsite	01/01/23	20
202	UI Designer	Full-time	Onsite	08/02/23	20
209	Accountant	Full-time	Onsite	01/01/23	10
210	Backend Engineer	Full-time	Online	07/22/23	60

**Bottom Window:**

- Query:** `/*select all from the view to check the result*/  
SELECT * FROM CheckBgnFinalRound;  
GO`
- Results:** A table titled "Results" showing three rows of data:

full_name	Background
Dongze Xie	Master in CS of SU
Jasper Xie	PhD in CS of MIT
Taeyeon Kim	Master in Statistics of Nanchang University

## Section C

test\_view3.sql - DESKTOP-S6ILFL9\SQLEXPRESS.HRDB (DESKTOP-S6ILFL9\wang (61)) - Microsoft SQL Server Management Studio

```
USE HRDB;
GO
/*select all to show the result of the view*/
SELECT * FROM CollectOnboardDoc;
```

Results (7 rows)

candidate full name	Type	URL
Dongze Xie	Disability Doc with Signature	url://dxie03_selfidentif789.pdf
Dongze Xie	Email	dxie03@yr.edu
Dongze Xie	Required Files	url://dxie03aaa_package.zip
Dongze Xie	Resume	url://dxie03_resume9892.pdf
Elon Musk	Cover Letter	url://elomus444ab_coverletter.docx
Elon Musk	Email	elon@yr.edu
Eren Yeager	Email	ailun@gmail.com

Query executed successfully.

SQLQuery2.sql - DE...S6ILFL9\wang (61)\* - Microsoft SQL Server Management Studio

```
USE HRDB;
GO
/*select all to show the result of the view*/
SELECT * FROM ListTestTimeDuration;
```

Results (9 rows)

TestID	time duration	Grade
4011	150 minutes	100
4021	120 minutes	98
4022	90 minutes	90
4023	90 minutes	92
4051	120 minutes	99
4052	120 minutes	99
4053	120 minutes	99
4071	150 minutes	65
4091	150 minutes	50

Query executed successfully.

## 2. Test User defined functions

### 2.1 Test code

```
testfn1-----  
-----  
USE HRDB;  
GO  
  
/*input candidateID to get this candidate's Resume*/  
PRINT dxdb.fnGetResume(8);  
PRINT ' ';
```

## Section C

```
PRINT dxdb.fnGetResume(3);
PRINT '';
PRINT dxdb.fnGetResume(1);

testfn2-----
USE HRDB;
GO
/**/
SELECT * FROM dxdb.fnGetInterviewerFeedback(609);
GO

SELECT * FROM dxdb.fnGetInterviewerFeedback(601);
GO

testfn3-----
USE HRDB;
GO

/*select all to show the return table*/
SELECT * FROM dxdb.fnListPostAfter('2022-6-15');
GO

testfn4-----
USE HRDB;
GO

PRINT dxdb.fnGetMostRemain();
GO

/*print which jobopening(ID) has the most remaining positions*/
```

## Section C

### 2.2 Test Result

The image shows two screenshots of Microsoft SQL Server Management Studio (SSMS) side-by-side.

**Screenshot 1 (test\_fn1.sql):** This screenshot shows the execution of a script named `test_fn1.sql`. The code uses the `fnGetResume` function to print resume details for candidate IDs 8, 3, and 1. The output window shows the results of the execution:

```
USE HRDB;
GO

/*input candidateID to get this candidate's Resume*/
PRINT dxdb.fnGetResume(8);
PRINT '';
PRINT dxdb.fnGetResume(3);
PRINT '';
PRINT dxdb.fnGetResume(1);

110 %  Messages
url://ty99988abc_kty.pdf
Not uploaded yet!
url//dxie03_resume9882.pdf

Completion time: 2022-12-06T23:57:47.6736328-05:00
```

**Screenshot 2 (test\_fn2.sql):** This screenshot shows the execution of a script named `test_fn2.sql`. The code uses the `fnGetInterviewerFeedback` function to select feedback from interviewers. The output window shows the results of the execution:

```
USE HRDB;
GO
SELECT * FROM dxdb.fnGetInterviewerFeedback(609);
GO

SELECT * FROM dxdb.fnGetInterviewerFeedback(601);
GO
```

Interviewer Name	date of interview	CandidateComment
Mrs Elena	2022-11-01	NULL
Mrs Elena	2022-11-01	I like this interviewer!
Mrs Elena	2022-11-01	I think the questions are not common and even m...
Mrs Elena	2022-11-01	The interviewer has strong accent, we can barel...
Mrs Elena	2022-11-05	NULL
Mrs Elena	2022-11-05	I like this interviewer!
Mrs Elena	2022-11-05	The interviewer has strong accent, we can barel...
Mrs Elena	2022-11-20	NULL

Interviewer Name	date of interview	CandidateComment
Roger Chen	2022-11-05	NULL
Roger Chen	2022-11-05	Nice experience!
Roger Chen	2022-11-05	Overall good
Roger Chen	2022-11-11	NULL
Roger Chen	2022-11-11	Nice experience!
Roger Chen	2022-11-11	Overall good

## Section C

The screenshot shows the Microsoft SQL Server Management Studio interface. The title bar indicates the connection is to DESKTOP-S6ILFL9\SQLEXPRESS.HRDB (DESKTOP-S6ILFL9\wang (57)). The Object Explorer pane shows the database structure, including the HRDB database. The script pane contains the following T-SQL code:

```
USE HRDB;
GO

/*select all to show the return table*/
SELECT * FROM dxdb.fnListPostAfter('2022-6-15');
GO
```

The results pane displays a table with three columns: OpeningID, JobPlatform, and PostDate. The data is as follows:

OpeningID	JobPlatform	PostDate
1	Indeed	2022-06-18 00:00:00.000
2	LinkedIn	2022-07-01 00:00:00.000
3	202	2022-07-01 00:00:00.000
4	Company Mainpage	2022-07-01 00:00:00.000
5	Indeed	2022-07-01 00:00:00.000
6	LinkedIn	2022-06-20 00:00:00.000
7	204	2022-06-20 00:00:00.000
8	Simplified Jobs	2022-06-18 00:00:00.000
9	Company Mainpage	2022-06-18 00:00:00.000
10	LinkedIn	2022-07-01 00:00:00.000

The status bar at the bottom shows "Query executed successfully." and "10 rows".

The screenshot shows the Microsoft SQL Server Management Studio interface. The title bar indicates the connection is to DESKTOP-S6ILFL9\SQLEXPRESS.HRDB (DESKTOP-S6ILFL9\wang (58)). The Object Explorer pane shows the database structure, including the HRDB database. The script pane contains the following T-SQL code:

```
USE HRDB;
GO

PRINT dxdb.fnGetMostRemain();
GO
```

The results pane displays a single row with the value 210. The status bar at the bottom shows "Query executed successfully." and "0 rows".

## 3. Test Stored Procedures

### 3.1 Test Code

```
testsp1-----  
-----  
USE HRDB;  
GO  
  
/*execute the stored procedure*/  
DECLARE @MyApplyTotal int;  
EXEC dxdb.spApplyTotal @OpeningID = 210, @ApplyTotal = @MyApplyTotal OUTPUT;
```

## Section C

```
DECLARE @MyApplyTotal209 int;
EXEC dxdb.spApplyTotal @OpeningID = 209, @ApplyTotal = @MyApplyTotal209 OUTPUT;

PRINT @MyApplyTotal;
PRINT @MyApplyTotal209;

testsp2tttt-----
/*add a new interviewer*/
USE HRDB;
GO

EXEC dxdb.spAddInterviewer
611, 'Mckenna Grace', 'Junior Software Engineer', 'Software Development';
GO

SELECT * FROM dxdb.Interviewers

testsp2fffffffffffff-----
USE HRDB;
GO

/*try to add an existing interviewer*/
BEGIN TRY
EXEC dxdb.spAddInterviewer
604, 'Sam Smith', 'Leader of Sales team', 'Product Marketing';
END TRY

BEGIN CATCH
PRINT 'An error occurred.';
PRINT 'Error Number: ' + CONVERT(VARCHAR, ERROR_NUMBER());
PRINT 'Message: ' + CONVERT(VARCHAR, ERROR_MESSAGE());
END CATCH
GO

testsp3-----
```

## Section C

```
USE HRDB;
GO

BEGIN TRY
EXEC dxdb.spUpdateGrade 4201,88;
END TRY

BEGIN CATCH
PRINT 'An error occurred.';
PRINT 'Error Number: ' + CONVERT(VARCHAR,ERROR_NUMBER());
PRINT 'Message: ' + CONVERT(VARCHAR, ERROR_MESSAGE());
END CATCH
GO

EXEC dxdb.spUpdateGrade 4101,88;
GO
SELECT * FROM dxdb.Tests WHERE TestID = 4101;
GO

testsp4-----
USE HRDB;
GO

BEGIN TRY /*test an online interview input*/
EXEC dxdb.spTrackOnsiteInterview 7402;
END TRY

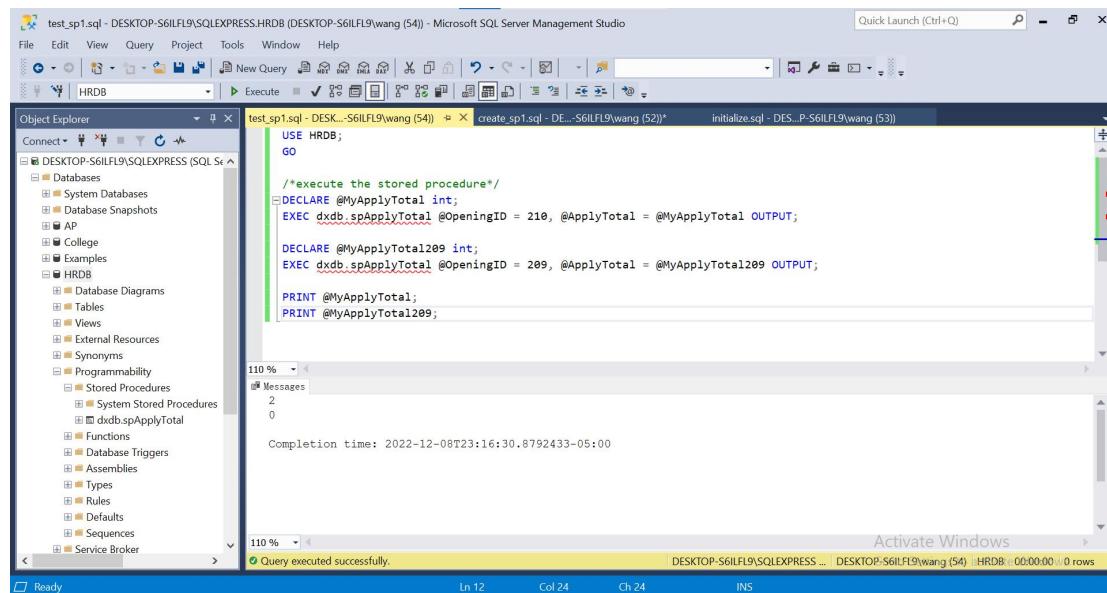
BEGIN CATCH
PRINT 'An error occurred.';
PRINT 'Error Number: ' + CONVERT(VARCHAR,ERROR_NUMBER());
PRINT 'Message: ' + CONVERT(VARCHAR, ERROR_MESSAGE());
END CATCH
GO

EXEC dxdb.spTrackOnsiteInterview 403;
```

## Section C

GO

### 3.2 Test Result



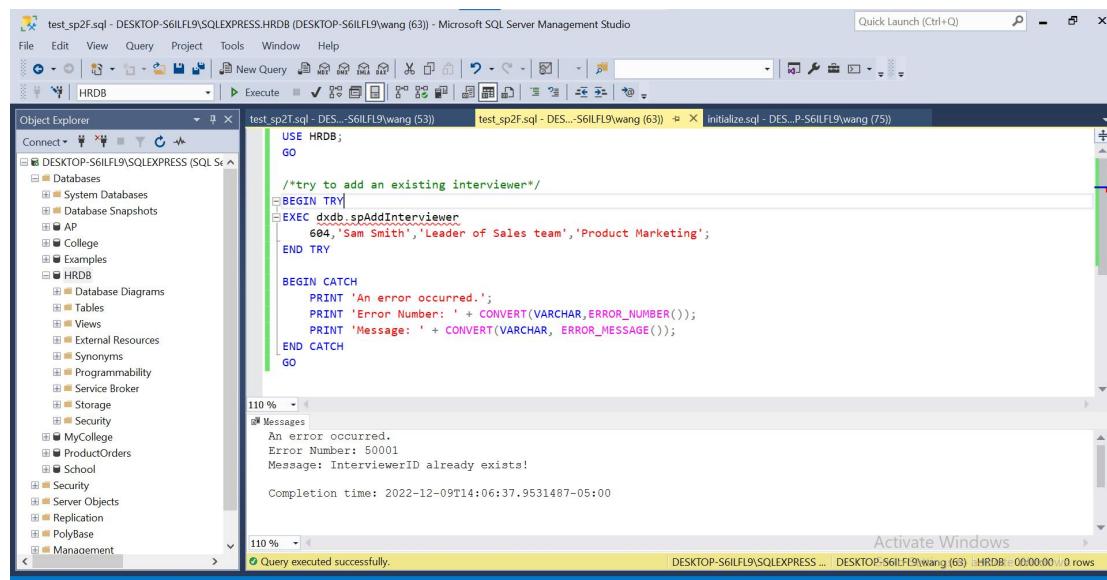
```
test_sp1.sql - DESKTOP-S6ILFL9\SQLEXPRESS.HRDB [DESKTOP-S6ILFL9\wang (54)] - Microsoft SQL Server Management Studio
File Edit View Query Project Tools Window Help
Object Explorer
Connect ▾ New Query Execute Find Replace Home Back Forward Stop Refresh
HRDB
test_sp1.sql - DESKTOP-S6ILFL9\SQLEXPRESS.HRDB [DESKTOP-S6ILFL9\wang (54)] | create_sp1.sql - DESKTOP-S6ILFL9\wang (52) | initialize.sql - DESKTOP-S6ILFL9\wang (53)
USE HRDB;
GO

/*execute the stored procedure*/
DECLARE @MyApplyTotal int;
EXEC dxdb.spApplyTotal @OpeningID = 210, @ApplyTotal = @MyApplyTotal OUTPUT;

DECLARE @MyApplyTotal209 int;
EXEC dxdb.spApplyTotal @OpeningID = 209, @ApplyTotal = @MyApplyTotal209 OUTPUT;

PRINT @MyApplyTotal;
PRINT @MyApplyTotal209;

Completion time: 2022-12-08T23:16:30.8792433-05:00
Messages
2
0
Query executed successfully.
```



```
test_sp2T.sql - DESKTOP-S6ILFL9\SQLEXPRESS.HRDB [DESKTOP-S6ILFL9\wang (63)] - Microsoft SQL Server Management Studio
File Edit View Query Project Tools Window Help
Object Explorer
Connect ▾ New Query Execute Find Replace Home Back Forward Stop Refresh
HRDB
test_sp2T.sql - DESKTOP-S6ILFL9\SQLEXPRESS.HRDB [DESKTOP-S6ILFL9\wang (63)] | test_sp2F.sql - DESKTOP-S6ILFL9\wang (63) | initialize.sql - DESKTOP-S6ILFL9\wang (75)
USE HRDB;
GO

/*try to add an existing interviewer*/
BEGIN TRY
    EXEC dxdb.spAddInterviewer
        604, 'Sam Smith', 'Leader of Sales team', 'Product Marketing';
END TRY

BEGIN CATCH
    PRINT 'An error occurred.';
    PRINT 'Error Number: ' + CONVERT(VARCHAR,ERROR_NUMBER());
    PRINT 'Message: ' + CONVERT(VARCHAR, ERROR_MESSAGE());
END CATCH
GO

Completion time: 2022-12-09T14:06:37.9531487-05:00
Messages
An error occurred.
Error Number: 50001
Message: InterviewerID already exists!
Query executed successfully.
```

## Section C

test\_sp2T.sql - DESKTOP-S6ILFL9\SQLEXPRESS.HRDB (DESKTOP-S6ILFL9\wang (53)) - Microsoft SQL Server Management Studio

```

USE HRDB;
GO

EXEC dxdb.spAddInterviewer
    611, 'Mcenna Grace', 'Junior Software Engineer', 'Software Development';
GO

SELECT * FROM dxdb.Interviewers

```

Results (11 rows)

InterviewerID	Name	Title	Department
1	Roger Chen	Head of Tech team	Software Development
2	Mr Lee	Java Consultant	Software Development
3	Dr Katarina	Product Manager	Software Development
4	Sam Smith	Leader of Sales team	Product Marketing
5	Dongze Real	CEO	Human Resource
6	Miss Sara	Leader of HR team	Human Resource
7	Andrew Sheeran	NULL	Human Resource
8	John Snow	NULL	Design and Art
9	Mrs Elena	Senior Software Engineer	Software Development
10	Dr Xie	Head of Finance Department	Finance
11	Mcenna Grace	Junior Software Engineer	Software Development

Query executed successfully.

test\_sp3.sql - DESKTOP-S6ILFL9\SQLEXPRESS.HRDB (DESKTOP-S6ILFL9\wang (52)) - Microsoft SQL Server Management Studio

```

USE HRDB;
GO

BEGIN TRY
    EXEC dxdb.spUpdateGrade 4201, 88;
END TRY

BEGIN CATCH
    PRINT 'An error occurred.';
    PRINT 'Error Number: ' + CONVERT(VARCHAR, ERROR_NUMBER());
    PRINT 'Message: ' + CONVERT(VARCHAR, ERROR_MESSAGE());
END CATCH
GO

EXEC dxdb.spUpdateGrade 4101, 88;
GO
SELECT * FROM dxdb.Tests WHERE TestID = 4101;
GO

```

Results (1 row affected)

An error occurred.  
Error Number: 50002  
Message: Not such TestID exists!

(1 row affected)

Query executed successfully.

test\_sp3.sql - DESKTOP-S6ILFL9\SQLEXPRESS.HRDB (DESKTOP-S6ILFL9\wang (52)) - Microsoft SQL Server Management Studio

```

USE HRDB;
GO

BEGIN TRY
    EXEC dxdb.spUpdateGrade 4201, 88;
END TRY

BEGIN CATCH
    PRINT 'An error occurred.';
    PRINT 'Error Number: ' + CONVERT(VARCHAR, ERROR_NUMBER());
    PRINT 'Message: ' + CONVERT(VARCHAR, ERROR_MESSAGE());
END CATCH
GO

EXEC dxdb.spUpdateGrade 4101, 88;
GO
SELECT * FROM dxdb.Tests WHERE TestID = 4101;
GO

```

Results (1 row affected)

TestID	InterviewerID	InterviewerID	StartTime	EndTime	Grade	Result
1	4101	9410	2022-11-05 13:00:00.000	2022-11-05 15:00:00.000	88	Pending

Query executed successfully.

## Section C

The screenshot shows the Microsoft SQL Server Management Studio interface. A query window titled 'test\_sp4.sql' is open, displaying the following T-SQL code:

```
USE HRDB;
GO

BEGIN TRY /*test an online interview input*/
EXEC dxdb.spTrackOnsiteInterview 7402;
END TRY

BEGIN CATCH
PRINT 'An error occurred.';
PRINT 'Error Number: ' + CONVERT(VARCHAR,ERROR_NUMBER());
PRINT 'Message: ' + CONVERT(VARCHAR, ERROR_MESSAGE());
END CATCH
GO

EXEC dxdb.spTrackOnsiteInterview 403;
GO
```

The results pane shows the output of the script execution:

Completion time:	2022-12-10T01:25:57.3254940-05:00
Query executed successfully.	DESKTOP-S6ILFL9\SQLEXPRESS ... DESKTOP-S6ILFL9\wang (59)   HRDB   00:00:00   1 rows

The screenshot shows the Microsoft SQL Server Management Studio interface. A query window titled 'test\_sp4.sql' is open, displaying the same T-SQL code as the previous screenshot:

```
USE HRDB;
GO

BEGIN TRY /*test an online interview input*/
EXEC dxdb.spTrackOnsiteInterview 7402;
END TRY

BEGIN CATCH
PRINT 'An error occurred.';
PRINT 'Error Number: ' + CONVERT(VARCHAR,ERROR_NUMBER());
PRINT 'Message: ' + CONVERT(VARCHAR, ERROR_MESSAGE());
END CATCH
GO

EXEC dxdb.spTrackOnsiteInterview 403;
GO
```

The results pane shows the output of the script execution, including a table with five columns: InterviewID, Airline reservation detail, Hotel reservation detail, Car rental detail, Receipt, and Expense Total. The table has one row with values: 403, https://www.jetblue.com/airlineinfo#aptx4869, https://www.airbnb.com/rooms#qwer666, https://www.hertz.com/rentacar/info#123456, url//receipt#aptx4869.pdf, and 850.00.

## 4. Test Transactions

### 4.1 Test Code

testtran1-----

```
USE HRDB;
```

```
GO
```

```
/*test the newly added candidate info*/
```

## Section C

```
SELECT * FROM dxdb.Candidates JOIN dxdb.CandidateInfo  
ON dxdb.Candidates.CandidateID = dxdb.CandidateInfo.CandidateID  
GO
```

```
SELECT * FROM dxdb.Documents  
GO
```

**testtran2-----**

```
USE HRDB;  
GO
```

```
/*check if truly deleted*/
```

```
SELECT * FROM dxdb.Candidates JOIN dxdb.CandidateInfo  
ON dxdb.Candidates.CandidateID = dxdb.CandidateInfo.CandidateID  
GO
```

```
SELECT * FROM dxdb.Documents
```

```
GO
```

**test-tran3-----**

```
USE HRDB;  
GO
```

```
/*test if tran3 indeed delete interviewer 603 */
```

```
SELECT * FROM dxdb.Interviewers;
```

**test\_tran4-----**

```
USE HRDB;  
GO
```

```
/*test if application 310 is processed and arranged interview&test by tran4*/
```

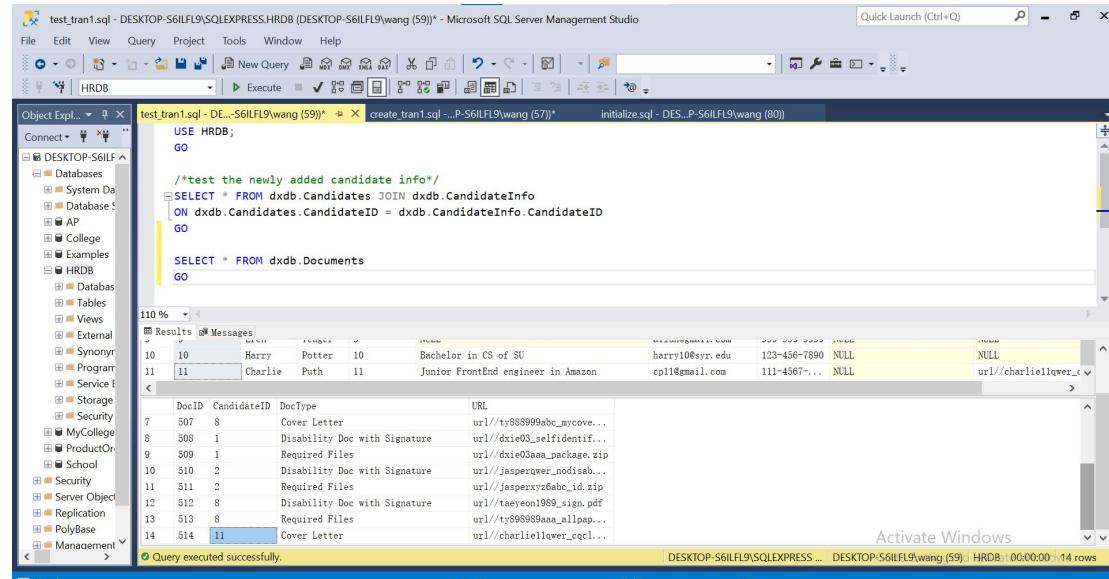
```
SELECT * FROM  
dxdb.Applications JOIN dxdb.Interviews  
ON dxdb.Applications.ApplicationID = dxdb.Interviews.ApplicationID  
JOIN dxdb.Tests
```

## Section C

```
ON dxdb.Tests.InterviewID = dxdb.Interviews.InterviewID
```

```
WHERE dxdb.Applications.ApplicationID = 310;
```

## 4.2 Test Result

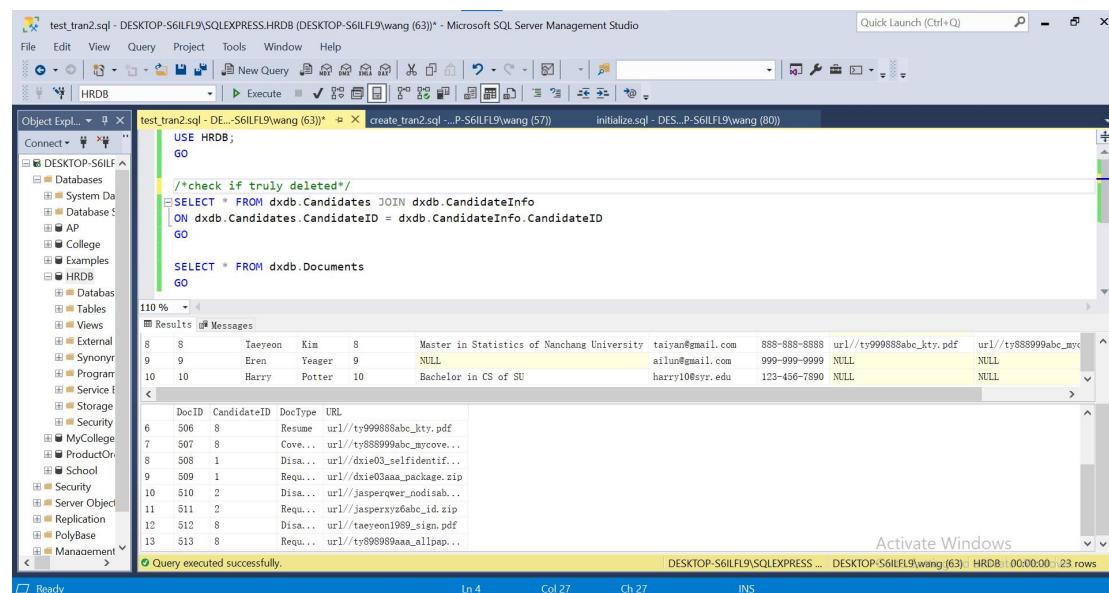


```
USE HRDB;
GO

/*test the newly added candidate info*/
SELECT * FROM dxdb.Candidates JOIN dxdb.CandidateInfo
ON dxdb.Candidates.CandidateID = dxdb.CandidateInfo.CandidateID
GO

SELECT * FROM dxdb.Documents
GO
```

DocID	CandidateID	DocType	URL
7	507	8	url/ty88899abc_myco...
8	508	1	Disability Doc with Signature
9	509	1	Required Files
10	510	2	Disability Doc with Signature
11	511	2	Required Files
12	512	8	Disability Doc with Signature
13	513	8	Required Files
14	514	11	Cover Letter



```
USE HRDB;
GO

/*check if truly deleted*/
SELECT * FROM dxdb.Candidates JOIN dxdb.CandidateInfo
ON dxdb.Candidates.CandidateID = dxdb.CandidateInfo.CandidateID
GO

SELECT * FROM dxdb.Documents
GO
```

DocID	CandidateID	DocType	URL
6	506	8	Resume
7	507	8	Cove...
8	508	1	Disa...
9	509	1	Requ...
10	510	2	Disa...
11	511	2	Requ...
12	512	8	Disa...
13	513	8	Requ...

## Section C

test\_tran3.sql - DESKTOP-S6ILFL9\SQLEXPRESS.HRDB (DESKTOP-S6ILFL9\wang (59)) - Microsoft SQL Server Management Studio

File Edit View Query Project Tools Window Help

HRDB

Object Explorer

test\_tran3.sql - DE...-S6ILFL9\wang (59) create\_tran3.sql - P-S6ILFL9\wang (57) initialize.sql - DES...P-S6ILFL9\wang (80)

```
USE HRDB;
GO

/*test if tran3 indeed delete interviewer 603 */
SELECT * FROM dxdb.Interviewers;
```

Results (0 Messages)

	InterviewerID	Name	Title	Department
1	601	Roger Chen	Head of Tech team	Software Development
2	602	Mr Lee	Java Consultant	Software Development
3	604	Sam Smith	Leader of Sales team	Product Marketing
4	605	Dongze Real	CEO	Human Resource
5	606	Miss Sara	Leader of HR team	Human Resource
6	607	Andrew Sheeran	NULL	Human Resource
7	608	John Snow	NULL	Design and Art
8	609	Mrs Elena	Senior Software Engineer	Software Development
9	610	Dr Xie	Head of Finance Department	Finance
10	611	Mckenna Grace	Junior Software Engineer	Software Development

Query executed successfully.

DESKTOP-S6ILFL9\SQLEXPRESS ... DESKTOP-S6ILFL9\wang (59) : HRDB : 00:00:00 | 10 rows

Activate Windows

test\_tran4.sql - DESKTOP-S6ILFL9\SQLEXPRESS.HRDB (DESKTOP-S6ILFL9\wang (51)) - Microsoft SQL Server Management Studio

File Edit View Query Project Tools Window Help

HRDB

Object Explorer

test\_tran4.sql - DE...-S6ILFL9\wang (51) create\_tran4.sql - P-S6ILFL9\wang (57) initialize.sql - DES...P-S6ILFL9\wang (80)

```
USE HRDB;
GO

/*test if application 310 is processed and arranged interview&test by tran4*/
SELECT *
FROM
    dxdb.Applications JOIN dxdb.Interviews
    ON dxdb.Applications.ApplicationID = dxdb.Interviews.ApplicationID
    JOIN dxdb.Tests
    ON dxdb.Tests.InterviewID = dxdb.Interviews.InterviewID
WHERE dxdb.Applications.ApplicationID = 310;
```

Results (0 Messages)

	ApplicationID	CandidateID	OpeningID	ApplyDate	Status	InterviewID	ApplicationID	Type	StartTime	EndTime	InterviewerID	Int
1	310	7	204	2022-10-01 00:00:00.000	On progress	411	310	Online	2022-11-23 13:00:00.000	2022-11-23 16:00:00.000	606	1

Query executed successfully.

DESKTOP-S6ILFL9\SQLEXPRESS ... DESKTOP-S6ILFL9\wang (51) : HRDB : 00:00:00 | 1 rows

Activate Windows

## Section C

### 5. Test Trigger

#### 5.1 Test Code

```
testtrigger1-----
USE HRDB;
GO
BEGIN TRAN
    SELECT NumOfPositions FROM dxdb.JobOpenings WHERE OpeningID = 204

    /*test the trigger by directly change pending application status into offer
extended*/
    UPDATE dxdb.Applications SET Status = 'Offer extended' WHERE ApplicationID = 310

    /*the number of remaining positions should reduced by 1*/
    SELECT NumOfPositions FROM dxdb.JobOpenings WHERE OpeningID = 204
COMMIT TRAN;

testtrigger2-----
USE HRDB;
GO

BEGIN TRAN
    INSERT INTO dxdb.Offers(OfferID,OpeningID,CandidateID,Status) VALUES
    (4,204,7,'Pending') /*since during last test for trigger we extend an offer to
application310*/

    UPDATE dxdb.StatusList SET Status = 'Offer extended' WHERE CandidateID = 7

    SELECT NumOfPositions FROM dxdb.JobOpenings WHERE OpeningID = 204

    UPDATE dxdb.Offers SET Status = 'Declined' WHERE OfferID = 4
    /*decline the offer to test trigRestoreNumOfPosition*/

    UPDATE dxdb.StatusList SET Status = 'On-call for next job opportunity' WHERE
```

## Section C

```
CandidateID = 7
```

```
SELECT NumOfPositions FROM dxdb.JobOpenings WHERE OpeningID = 204  
COMMIT TRAN
```

```
testtrigger3-----
```

```
USE HRDB;  
GO
```

```
BEGIN TRAN
```

```
SELECT * FROM dxdb.StatusList WHERE CandidateID = 10
```

```
INSERT INTO dxdb.Applications  
(ApplicationID,CandidateID,OpeningID,ApplyDate,Status) VALUES  
(311, 10, 210, '2022-10-12', 'Pending')
```

```
SELECT * FROM dxdb.StatusList WHERE CandidateID = 10
```

```
/*check if trigReApply success*/
```

```
COMMIT TRAN;
```

```
testtrigger4-----
```

```
USE HRDB;  
GO
```

```
BEGIN TRAN
```

```
SELECT Status FROM dxdb.Applications WHERE ApplicationID = 301  
/*change the result from handling into invalid to test trigInvalidComplaint*/  
UPDATE dxdb.Complaints SET Result = 'Invalid' WHERE ComplaintID = 1
```

```
SELECT Status FROM dxdb.Applications WHERE ApplicationID = 301
```

```
COMMIT TRAN;
```

## 5.2 Test Result

## Section C

**test\_trig1.sql - DESKTOP-S6ILFL9\SQLEXPRESS.HRDB (DESKTOP-S6ILFL9\wang (65))\* - Microsoft SQL Server Management Studio**

```
USE HRDB;
GO
BEGIN TRAN
    SELECT NumOfPositions FROM dxdb.JobOpenings WHERE OpeningID = 204

    /*test the trigger by directly change pending application status into offer extended*/
    UPDATE dxdb.Applications SET Status = 'Offer extended' WHERE ApplicationID = 310

    /*the number of remaining positions should reduced by 1*/
    SELECT NumOfPositions FROM dxdb.JobOpenings WHERE OpeningID = 204
COMMIT TRAN;
```

110 %

Results  Messages

NumOfPositions	
1	5
1	4

Query executed successfully.

**test\_trig2.sql - DESKTOP-S6ILFL9\SQLEXPRESS.HRDB (DESKTOP-S6ILFL9\wang (56))\* - Microsoft SQL Server Management Studio**

```
BEGIN TRAN
    INSERT INTO dxdb.Offers(OfferID,OpeningID,CandidateID,Status) VALUES
    (4,204,7, 'Pending') /*since during last test for trigger we extend an offer to application310*/

    UPDATE dxdb.StatusList SET Status = 'Offer extended' WHERE CandidateID = 7

    SELECT NumOfPositions FROM dxdb.JobOpenings WHERE OpeningID = 204

    UPDATE dxdb.Offers SET Status = 'Declined' WHERE OfferID = 4
    /*decline the offer to test trigRestoreNumOfPosition*/

    UPDATE dxdb.StatusList SET Status = 'On-call for next job opportunity' WHERE CandidateID = 7

    SELECT NumOfPositions FROM dxdb.JobOpenings WHERE OpeningID = 204
COMMIT TRAN;
```

110 %

Results  Messages

NumOfPositions	
1	4
1	5

Query executed successfully.

**test\_trig3.sql - DESKTOP-S6ILFL9\SQLEXPRESS.HRDB (DESKTOP-S6ILFL9\wang (53))\* - Microsoft SQL Server Management Studio**

```
USE HRDB;
GO
BEGIN TRAN
    SELECT * FROM dxdb.StatusList WHERE CandidateID = 10

    INSERT INTO dxdb.Applications (ApplicationID,CandidateID,OpeningID,ApplyDate,Status) VALUES
    (311, 10, 210, '2022-10-12', 'Pending')

    SELECT * FROM dxdb.StatusList WHERE CandidateID = 10
    /*check if trigReApply success*/
COMMIT TRAN;
```

110 %

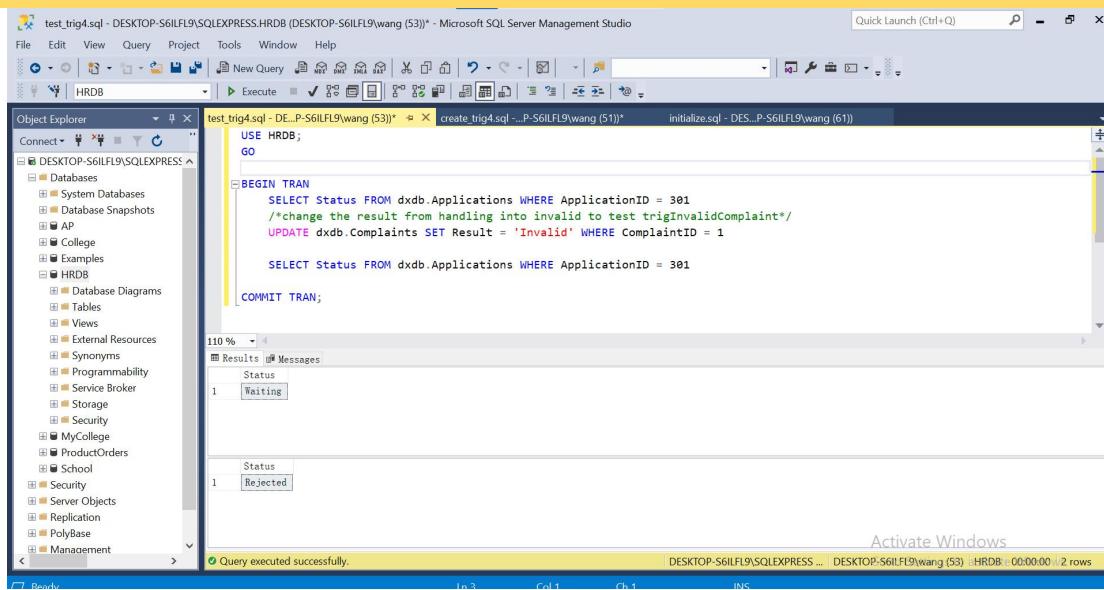
Results  Messages

CandidateID	FullName	Status
10	Harry Potter	Rejected

CandidateID	FullName	Status	
1	10	Harry Potter	Applied

Query executed successfully.

## Section C



The screenshot shows the Microsoft SQL Server Management Studio interface. The title bar reads "test\_trig4.sql - DESKTOP-S6ILFL9\SQLEXPRESS.HRDB (DESKTOP-S6ILFL9\wang (53))\* - Microsoft SQL Server Management Studio". The menu bar includes File, Edit, View, Query, Project, Tools, Window, Help. The toolbar has various icons for file operations like New Query, Save, Print, and Execute. The Object Explorer on the left shows the database structure under "DESKTOP-S6ILFL9\SQLEXPRESS.HRDB", including Databases, Tables, Views, and Security. The main pane contains a query window with the following T-SQL code:

```
USE HRDB;
GO

BEGIN TRAN
    SELECT Status FROM dxdb.Applications WHERE ApplicationID = 301
    /*change the result from handling into invalid to test trigInvalidComplaint*/
    UPDATE dxdb.Complaints SET Result = 'Invalid' WHERE ComplaintID = 1

    SELECT Status FROM dxdb.Applications WHERE ApplicationID = 301

COMMIT TRAN;
```

The results pane shows two rows of data:

Status
Waiting
Rejected

A status bar at the bottom indicates "Query executed successfully." and "DESKTOP-S6ILFL9\SQLEXPRESS... DESKTOP-S6ILFL9\wang (53) | HRDB | 00:00:00 | 2 rows".

## 6. Business Reports' test result is in Appendix

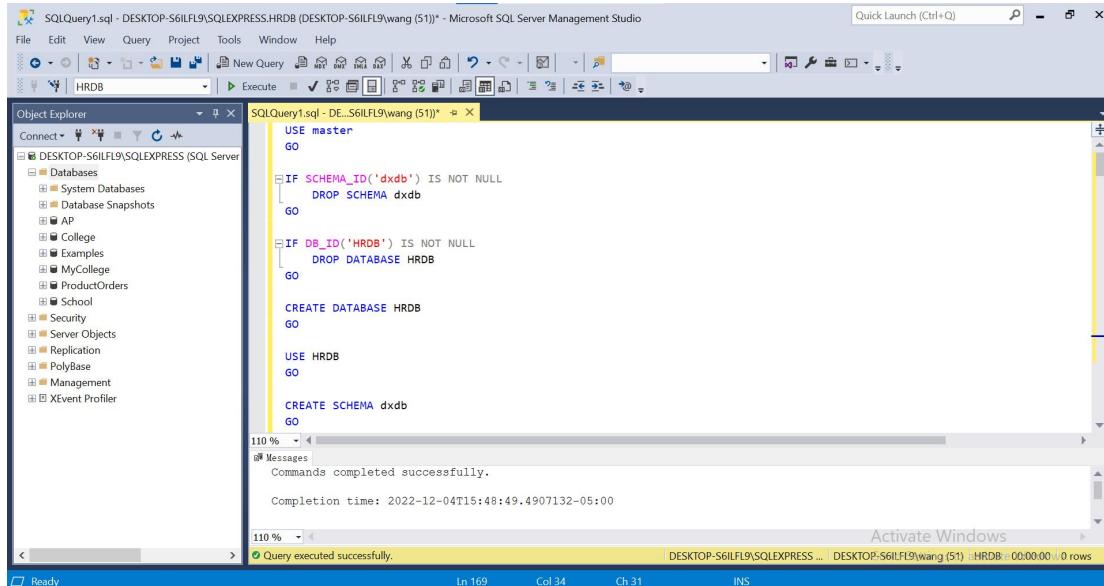
### Section D. Conclusion

This is definitely a comprehensive practice for this course. The load is overwhelming but also a valuable experience because of that. I'm proud that I covered the statements mentioned in project requirements and completed the HRDB. And also thanks the instructor and TAs for grading this long report.

# Appendix

## Appendix

### 1. Screenshots for database set up



The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left shows the connection to DESKTOP-S6ILFL9\SQLEXPRESS and the database HRDB. The central pane displays a T-SQL script for creating the database:

```
USE master
GO

IF SCHEMA_ID('dxdb') IS NOT NULL
    DROP SCHEMA dxdb
GO

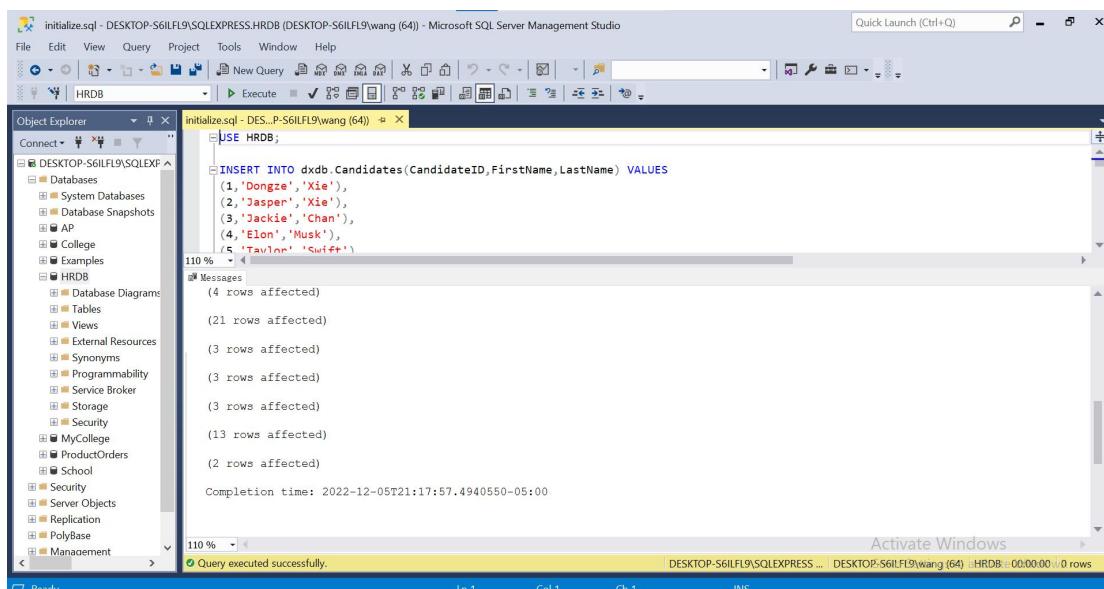
IF DB_ID('HRDB') IS NOT NULL
    DROP DATABASE HRDB
GO

CREATE DATABASE HRDB
GO

USE HRDB
GO

CREATE SCHEMA dxdb
GO
```

The status bar at the bottom indicates "Query executed successfully." and "Completion time: 2022-12-04T15:48:49.4907132-05:00".



The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left shows the connection to DESKTOP-S6ILFL9\SQLEXPRESS and the database HRDB. The central pane displays a T-SQL script for inserting data into the Candidates table:

```
USE HRDB;
GO

INSERT INTO dxdb.Candidates(CandidateID,FirstName,LastName) VALUES
(1,'Dongze','Xie'),
(2,'Jasper','Xie'),
(3,'Jackie','Chan'),
(4,'Elon','Musk'),
(5,'Taylor','Swift')
```

The status bar at the bottom indicates "Query executed successfully." and "Completion time: 2022-12-05T21:17:57.4940550-05:00".

### 2. Screenshots for create views

# Appendix

create\_view1.sql - DESKTOP-S6ILFL9\SQLEXPRESS.HRDB (DESKTOP-S6ILFL9\wang (65)) - Microsoft SQL Server Management Studio

```

File Edit View Query Project Tools Window Help
Object Explorer Connect New Query Execute
HRDB
  System Databases
  Database Snapshots
  AP
  College
  Examples
  HRDB
  Database Diagrams
  Tables
  Views
    System Views
      dbo.FindOpeningsInSyr
  External Resources
  Synonyms
  Programmability
  Service Broker
  Storage
  Security
    MyCollege
    ProductOrders
    School
  Security
  Server Objects
  Replication
  PolyBase
  Management
  Ready
  
```

```

test.view1.sql - DE...S6ILFL9\wang (52) create_view1.sql ...P-S6ILFL9\wang (65)* initialize.sql - DES...P-S6ILFL9\wang (63)
/*view1: find all job openings located in Syracuse*/
USE HRDB;
GO

IF OBJECT_ID('FindOpeningsInSyr') IS NOT NULL
DROP VIEW FindOpeningsInSyr
GO

CREATE VIEW FindOpeningsInSyr AS
SELECT OpeningID, Position, Type, Medium,
CONVERT(varchar, JobStartDate, 101) AS [Start Date],
NumOfPositions AS [Recent Openings]
FROM dxdb.Jobs JOIN dxdb.JobOpenings
ON dxdb.Jobs.JobID=dxdb.JobOpenings.JobID
WHERE Location='Syracuse'
GO

110 %
Messages
Commands completed successfully.

Completion time: 2022-12-05T21:53:54.0091626-05:00
110 %
Query executed successfully.
  
```

In 11 Col 40 Ch 34 INS

create\_view2.sql - DESKTOP-S6ILFL9\SQLEXPRESS.HRDB (DESKTOP-S6ILFL9\wang (65)) - Microsoft SQL Server Management Studio

```

File Edit View Query Project Tools Window Help
Object Explorer Connect New Query Execute
HRDB
  System Databases
  Database Snapshots
  AP
  College
  Examples
  HRDB
  Database Diagrams
  Tables
  Views
    System Views
      dbo.CheckBgnFinalRound
      dbo.FindOpeningsInSyr
  External Resources
  Synonyms
  Programmability
  Service Broker
  Storage
  Security
    MyCollege
    ProductOrders
    School
  Security
  Server Objects
  Replication
  PolyBase
  Management
  Ready
  
```

```

test.view2.sql - DE...S6ILFL9\wang (53) create_view2.sql ...P-S6ILFL9\wang (65)* initialize.sql - DES...P-S6ILFL9\wang (63)
/*view2: Check candidates' background who enter the final(3rd) interview*/
USE HRDB;
GO

IF OBJECT_ID('CheckBgnFinalRound') IS NOT NULL
DROP VIEW CheckBgnFinalRound
GO

CREATE VIEW CheckBgnFinalRound AS
SELECT FirstName+' '+LastName AS [full name], Background
FROM dxdb.Candidates JOIN dxdb.CandidateInfo
ON dxdb.Candidates.CandidateID = dxdb.CandidateInfo.CandidateID
JOIN dxdb.Applications ON dxdb.Applications.CandidateID= dxdb.Candidates.CandidateID
JOIN dxdb.Interviews ON dxdb.Interviews.ApplicationID = dxdb.Applications.ApplicationID
WHERE InterviewRound = 3;
GO

110 %
Messages
Commands completed successfully.

Completion time: 2022-12-05T23:32:04.6703422-05:00
110 %
Query executed successfully.
  
```

In 16 Col 3 Ch 3 INS

create\_view3.sql - DESKTOP-S6ILFL9\SQLEXPRESS.HRDB (DESKTOP-S6ILFL9\wang (64)) - Microsoft SQL Server Management Studio

```

File Edit View Query Project Tools Window Help
Object Explorer Connect New Query Execute
HRDB
  Database Snapshots
  AP
  College
  Examples
  HRDB
  Database Diagrams
  Tables
  Views
    System Views
      dbo.CheckBgnFinalRound
      dbo.CollectOnboardDoc
      dbo.FindOpeningsInSyr
  External Resources
  Synonyms
  Programmability
  Service Broker
  Storage
  Security
    MyCollege
    ProductOrders
    School
  Security
  Server Objects
  Replication
  PolyBase
  Management
  Ready
  
```

```

test.view3.sql - DE...S6ILFL9\wang (61) create_view3.sql ...P-S6ILFL9\wang (64)* SQLQuery2.sql - DE...S6ILFL9\wang (55)*
/*view3: Collect all documents from whom is under onboarding procedure*/
USE HRDB;
GO

IF OBJECT_ID('CollectOnboardDoc') IS NOT NULL
DROP VIEW CollectOnboardDoc
GO

CREATE VIEW CollectOnboardDoc AS
SELECT FirstName+' '+LastName AS [candidate full name], DocType AS Type, URL
FROM dxdb.Onboarding JOIN dxdb.Documents
ON dxdb.Onboarding.CandidateID = dxdb.Documents.CandidateID
JOIN dxdb.Candidates ON dxdb.Candidates.CandidateID = dxdb.Onboarding.CandidateID
UNION
SELECT FirstName+' '+LastName AS [candidate full name], 'Email' AS Type, Email AS URL
FROM dxdb.CandidateInfo JOIN dxdb.Onboarding
ON dxdb.CandidateInfo.CandidateID = dxdb.Onboarding.CandidateID
JOIN dxdb.Candidates ON dxdb.Candidates.CandidateID = dxdb.CandidateInfo.CandidateID
GO

110 %
Messages
Commands completed successfully.

Completion time: 2022-12-06T09:09:37.6445834-05:00
110 %
Query executed successfully.
  
```

In 18 Col 97 Ch 88 INS

# Appendix

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left shows the database structure for 'HRDB'. The central pane displays the following T-SQL code:

```
/*view: list all tests with their time duration which was in an completed online interview */
USE HRDB;
GO

IF OBJECT_ID('ListTestTimeDuration') IS NOT NULL
DROP VIEW ListTestTimeDuration
GO

CREATE VIEW ListTestTimeDuration AS
SELECT TestID,
       CAST(DATEDIFF(MINUTE,dxdb.Tests.StartTime,dxdb.Tests.EndTime) AS VARCHAR) +' minutes' AS [time duration],Grade
FROM dxdb.Tests JOIN dxdb.Interviews
ON dxdb.Tests.InterviewID = dxdb.Interviews.InterviewID
WHERE Type='Online' AND Result != 'Pending'
Go
```

The status bar at the bottom indicates 'Query executed successfully.'

## 3. Screenshots for create user defined functions

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left shows the database structure for 'HRDB'. The central pane displays the following T-SQL code:

```
/*fn: get the resume url of certain candidate*/
USE HRDB;
GO

IF OBJECT_ID('dxdb.fnGetResume')IS NOT NULL
DROP FUNCTION dxdb.fnGetResume
GO

CREATE FUNCTION dxdb.fnGetResume
    (@CandidateID int)
    RETURNS VARCHAR(50)
BEGIN
    RETURN(
        SELECT ISNULL(ResumeURL,'Not uploaded yet!') FROM dxdb.CandidateInfo
        WHERE CandidateID = @CandidateID
    );
END;
GO
```

The status bar at the bottom indicates 'Query executed successfully.'

# Appendix

create\_fn2.sql - DESKTOP-S6ILFL9\SQLEXPRESS.HRDB (DESKTOP-S6ILFL9\wang (52)) - Microsoft SQL Server Management Studio

```

Object Explorer
File Edit View Project Tools Window Help
Connect New Query Execute Quick Launch (Ctrl+Q) ×
HRDB
CREATE FUNCTION dxdb.fnGetInterviewerFeedBack
    (@InterviewerID int)
    RETURNS table
    RETURN(
        SELECT Name AS [Interviewer Name], CONVERT(date,StartTime) AS [date of interview], CandidateComment
        FROM dxdb.Interviewers JOIN dxdb.Evaluation
        ON dxdb.Interviewers.InterviewerID = dxdb.Evaluation.InterviewerID
        JOIN dxdb.Interviews ON dxdb.Interviews.InterviewerID = dxdb.Interviewers.InterviewerID
        WHERE dxdb.Interviewers.InterviewerID = @InterviewerID
    UNION
        SELECT Name AS [Interviewer Name], CONVERT(date,StartTime) AS [date of interview], Description
        FROM dxdb.Interviewers JOIN dxdb.Interviews
        ON dxdb.Interviews.InterviewerID = dxdb.Interviewers.InterviewerID
        JOIN dxdb.Complaints ON dxdb.Complaints.InterviewerID = dxdb.Interviews.InterviewerID
        WHERE dxdb.Interviewers.InterviewerID = @InterviewerID
    );

```

Completion time: 2022-12-07T00:42:46.7115269-05:00

Query executed successfully.

create\_fn3.sql - DESKTOP-S6ILFL9\SQLEXPRESS.HRDB (DESKTOP-S6ILFL9\wang (77)) - Microsoft SQL Server Management Studio

```

Object Explorer
File Edit View Query Project Tools Window Help
Connect New Query Execute Quick Launch (Ctrl+Q) ×
HRDB
/*fn3: List all the job opening posts after certain date*/
USE HRDB;
GO

IF OBJECT_ID('dxdb.fnListPostAfter') IS NOT NULL
DROP FUNCTION dxdb.fnListPostAfter
GO

CREATE FUNCTION dxdb.fnListPostAfter
    (@afterdate datetime)
    RETURNS table
    RETURN(
        SELECT * FROM dxdb.PlatformPosts
        WHERE PostDate > @afterdate
    );

```

Completion time: 2022-12-08T17:42:52.1670587-05:00

Query executed successfully.

create\_fn4.sql - DESKTOP-S6ILFL9\SQLEXPRESS.HRDB (DESKTOP-S6ILFL9\wang (52)) - Microsoft SQL Server Management Studio

```

Object Explorer
File Edit View Project Tools Window Help
Connect New Query Execute Quick Launch (Ctrl+Q) ×
HRDB
/*fn4: get the current job opening with the most remaining positions*/
USE HRDB;
GO

IF OBJECT_ID('dxdb.fnGetMostRemain') IS NOT NULL
DROP FUNCTION dxdb.fnGetMostRemain
GO

CREATE FUNCTION dxdb.fnGetMostRemain()
    RETURNS int
    BEGIN
        RETURN(SELECT OpeningID FROM dxdb.JobOpenings
        WHERE NumOfPositions =
            (SELECT MAX(NumOfPositions) FROM dxdb.JobOpenings));
    END;
GO

```

Completion time: 2022-12-08T22:31:47.5875994-05:00

Query executed successfully.

# Appendix

## 4. Screenshots for create sp

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left shows the database structure for HRDB. The main pane displays the following T-SQL code:

```
--sp1: check how many candidates applied a specific jobOpening
USE HRDB;
GO

IF OBJECT_ID('dxdbo.spApplyTotal') IS NOT NULL
DROP PROC dxdbo.spApplyTotal
GO

CREATE PROC dxdbo.spApplyTotal
@OpeningID int,
@ApplyTotal int OUTPUT
AS
SELECT @ApplyTotal = COUNT(*)
FROM dxdbo.Applications
WHERE OpeningID = @OpeningID
GO
```

The status bar at the bottom indicates "Query executed successfully."

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left shows the database structure for HRDB. The main pane displays the following T-SQL code:

```
--sp2: add a new interviewer and fill in his/her info
USE HRDB;
GO

IF OBJECT_ID('dxdbo.spAddInterviewer') IS NOT NULL
DROP PROC dxdbo.spAddInterviewer
GO

CREATE PROC dxdbo.spAddInterviewer
@InterviewerID int,
@Name varchar(50),
@Title varchar(50),
@Department varchar(50)
AS
IF EXISTS(SELECT * FROM dxdbo.Interviewers WHERE InterviewerID = @InterviewerID)
THROW 50001, 'InterviewerID already exists!',1;
ELSE
INSERT INTO dxdbo.Interviewers
VALUES (@InterviewerID, @Name, @Title, @Department);
```

The status bar at the bottom indicates "Query executed successfully."

# Appendix

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left shows a database named HRDB. The central pane displays the script for creating a stored procedure:

```
/*sp3 update the grade of a specific test*/
USE HRDB;
GO

IF OBJECT_ID('dxdb.spUpdateGrade') IS NOT NULL
DROP PROC dxdb.spUpdateGrade
GO

CREATE PROC dxdb.spUpdateGrade
@TestID int , @Grade int
AS
IF @TestID IS NULL OR (SELECT COUNT(*) FROM dxdb.Tests WHERE TestID = @TestID) = 0
    THROW 50002, 'Not such TestID exists!',1;
ELSE
    UPDATE dxdb.Tests SET Grade = @Grade WHERE TestID = @TestID
GO
```

The status bar at the bottom indicates "Query executed successfully."

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left shows a database named HRDB. The central pane displays the script for creating a stored procedure:

```
/*sp4: track a specific onsite interview*/
USE HRDB;
GO

IF OBJECT_ID('dxdb.spTrackOnsiteInterview') IS NOT NULL
DROP PROC dxdb.spTrackOnsiteInterview
GO

CREATE PROC dxdb.spTrackOnsiteInterview
@InterviewID int
AS
IF (SELECT Type FROM dxdb.Interviews WHERE InterviewID = @InterviewID)='Online'
    THROW 50003, 'This is an online interview!',1;
ELSE
    SELECT dxdb.TripTracking.InterviewID, AirlineDetail AS [Airline reservation detail],
    HotelDetail AS [Hotel reservation detail], CarRentalDetail AS [Car rental detail],
    Receipt, Amount AS [Expense Total]
    FROM dxdb.TripTracking JOIN dxdb.TripExpense
    ON dxdb.TripTracking.InterviewID = dxdb.TripExpense.InterviewID
    WHERE dxdb.TripTracking.InterviewID = @InterviewID
GO
```

The status bar at the bottom indicates "Query executed successfully."

## 5. Screenshots for create transactions

## Appendix

create\_tran1.sql - DESKTOP-S6ILFL9\SQLEXPRESS.HRDB (DESKTOP-S6ILFL9\wang (57)) - Microsoft SQL Server Management Studio

File Edit View Query Project Tools Window Help

New Query

Execute

Object Explorer

Connect

HRDB

test\_tran1.sql - DE... -S6ILFL9\wang (59) create\_tran1.sql - P-S6ILFL9\wang (57)\* initialize.sql - DES... -P-S6ILFL9\wang (80)

```
/*tran1: add a new candidate*/
USE HRDB;
GO

BEGIN TRAN
    INSERT INTO dxdb.Candidates (CandidateID,FirstName,LastName) VALUES
    (11,'Charlie','Puth');

    INSERT INTO dxdb.CandidateInfo(CandidateID,Background,Email,Phone,ResumeURL,CoverLetterURL)VALUES
    (11,'Junior FrontEnd engineer in Amazon','cp11@gmail.com','111-4567-1234',null,'url//charlie11qwer_cqcl.pdf');

    INSERT INTO dxdb.Documents(DocID,CandidateID,DocType,URL) VALUES
    (514,11,'Cover Letter','url//charlie11qwer_cqcl.pdf')

COMMIT TRAN
```

Completion time: 2022-12-10T21:59:29.6947653-05:00

Query executed successfully.

Activate Windows

create\_tran2.sql - DESKTOP-S6ILFL9\SQLEXPRESS.HRDB (DESKTOP-S6ILFL9\wang (57)) - Microsoft SQL Server Management Studio

File Edit View Query Project Tools Window Help

New Query

Execute

Object Explorer

Connect

HRDB

create\_tran2.sql - P-S6ILFL9\wang (57) initialize.sql - DES\_P-S6ILFL9\wang (80)

```
/*tran2: delete an inactive candidate(no application, without resume, not on the special list)*/
USE HRDB;
GO

BEGIN TRAN;
DELETE FROM dxdb.Documents WHERE CandidateID = 11;
DELETE FROM dxdb.CandidateInfo WHERE CandidateID = 11;
DELETE FROM dxdb.Candidates WHERE CandidateID = 11;

IF @@ROWCOUNT > 3
    BEGIN
        ROLLBACK TRAN;
        PRINT 'This affects more than expected, deletions rolled back!';
    END
ELSE

    BEGIN
        PRINT '(1 row affected)';
        PRINT '(1 row affected)';
        PRINT '(1 row affected)';
        Deletions committed to the database!
    END
```

Completion time: 2022-12-10T22:09:26.2084376-05:00

Query executed successfully.

Activate Windows

create\_tran3.sql - DESKTOP-S6ILFL9\SQLEXPRESS.HRDB (DESKTOP-S6ILFL9\wang (57)) - Microsoft SQL Server Management Studio

File Edit View Query Project Tools Window Help

New Query

Execute

Object Explorer

Connect

HRDB

create\_tran3.sql - P-S6ILFL9\wang (57) initialize.sql - DES\_P-S6ILFL9\wang (80)

```
/*tran3: delete an interviewer who has never participated any interview*/
USE HRDB;
GO

BEGIN TRAN;
IF (SELECT COUNT(*) FROM dxdb.Interviews WHERE InterviewerID = 603)=0
    AND (SELECT COUNT(*) FROM dxdb.Evaluation WHERE InterviewerID = 603)=0
    AND (SELECT COUNT(*) FROM dxdb.Tests WHERE InterviewerID = 603)=0
BEGIN
    DELETE FROM dxdb.Interviewers WHERE InterviewerID = 603;
    IF @@ROWCOUNT > 1
        BEGIN
            ROLLBACK TRAN;
            PRINT 'This affects more than one row! Deletion rolled back!';
        END
    END;
```

(1 row affected)  
Deletion success!

Completion time: 2022-12-10T23:36:31.0525987-05:00

Query executed successfully.

Activate Windows

# Appendix

A screenshot of Microsoft SQL Server Management Studio (SSMS) showing a query window titled 'create\_tran4.sql'. The code in the window is a transaction script:

```
/*tran4: process an application*/
USE HRDB;
GO

BEGIN TRAN
    UPDATE dxdb.Applications SET Status='On progress' WHERE ApplicationID = 310;

    INSERT INTO dxdb.Interviews (InterviewID,ApplicationID,Type,StartTime,EndTime,InterviewerID,InterviewRound) VALUES
        (411,310,'Online','2022-11-23 13:00:00','2022-11-23 16:00:00',606,1);

    INSERT INTO dxdb.Tests (TestID,InterviewID,InterviewerID,StartTime,EndTime,Grade,Result) VALUES
        (411,411,606,'2022-11-23 13:00:00','2022-11-23 15:40:00',0,'Pending');

    COMMIT TRAN
```

The 'Messages' pane shows three rows affected by the insert statements. The status bar at the bottom indicates 'Query executed successfully.'

## 6. Screenshots for create triggers

A screenshot of Microsoft SQL Server Management Studio (SSMS) showing a query window titled 'test\_trig1.sql'. The code in the window is a trigger creation script:

```
/*trig1: every time a position is filled by a candidate,
   the number of job openings for that position should be reduced by 1*/
USE HRDB;
GO

DROP TRIGGER IF EXISTS dxdb.trigReduceOpening;
GO

CREATE TRIGGER dxdb.trigReduceOpening
ON dxdb.Applications
AFTER UPDATE
AS
    IF (SELECT Status FROM inserted) = 'Offer extended'
        BEGIN
            UPDATE dxdb.JobOpenings
            SET NumOfPositions = NumOfPositions -
                (SELECT inserted.OpeningID
                 FROM dxdb.JobOpenings JOIN inserted ON dxdb.JobOpenings.OpeningID = inserted.OpeningID)
        END;
```

The 'Messages' pane shows the command completed successfully. The status bar at the bottom indicates 'Query executed successfully.'

# Appendix

The image displays three separate windows of Microsoft SQL Server Management Studio (SSMS) showing the creation of triggers for the HRDB database.

**Screenshot 1 (Top):** Shows the creation of trigger `trig2`. The code is as follows:

```

/*trig2: every time an offer is declined,
the corresponding job opening should increased by 1*/
USE HRDB;
GO

DROP TRIGGER IF EXISTS dxdb.trigRestoreNumOfPosition;
GO

CREATE TRIGGER dxdb.trigRestoreNumOfPosition
ON dxdb.Offers
AFTER UPDATE
AS
IF (SELECT Status FROM inserted) = 'Declined'
BEGIN
    UPDATE dxdb.JobOpenings
    SET NumOfPositions = NumOfPositions + 1
    WHERE dxdb.JobOpenings.OpeningID =
        (SELECT inserted.OpeningID
        FROM dxdb.JobOpenings JOIN inserted ON dxdb.JobOpenings.OpeningID = inserted.OpeningID)
END;
  
```

**Screenshot 2 (Middle):** Shows the creation of trigger `trig3`. The code is as follows:

```

DROP TRIGGER IF EXISTS dxdb.trigReApply;
GO

CREATE TRIGGER dxdb.trigReApply
ON dxdb.Applications
AFTER INSERT
AS
IF (SELECT Status FROM dxdb.Applications WHERE CandidateID =
    (SELECT CandidateID FROM inserted)
    AND OpeningID = (SELECT OpeningID FROM inserted)
    AND ApplicationID != (SELECT ApplicationID FROM inserted)) = 'Rejected'
/*Only take place when re-apply the same opening which has been rejected in the past*/
BEGIN
    UPDATE dxdb.StatusList SET Status='Applied'
    WHERE CandidateID = (SELECT CandidateID FROM inserted)
END;
  
```

**Screenshot 3 (Bottom):** Shows the creation of trigger `trig4`. The code is as follows:

```

/*trig4: After the Complaints department finds that the
complaint is invalid, status is changed to rejected */
USE HRDB;
GO

DROP TRIGGER IF EXISTS dxdb.trigInvalidComplaint;
GO

CREATE TRIGGER dxdb.trigInvalidComplaint
ON dxdb.Complaints
AFTER UPDATE
AS
IF (SELECT Result FROM inserted)='Invalid'
BEGIN
    UPDATE dxdb.Applications SET Status = 'Rejected'
    WHERE dxdb.Applications.ApplicationID =
        (SELECT ApplicationID FROM inserted)
END;
  
```

## 7. Screenshots for create scripts

# Appendix

The screenshot shows three separate instances of Microsoft SQL Server Management Studio (SSMS) running side-by-side, each displaying a different SQL script in the Query Editor tab.

- Top Window:** Title: "create\_roles.sql - DESKTOP-S6ILFL9\SQLEXPRESS.HRDB (DESKTOP-S6ILFL9\wang (51))". It contains scripts for creating roles InterviewAdmin and DocAdmin, and adding them to the db\_datareader role. The output window shows "Commands completed successfully." and a completion time of 2022-12-11T16:49:32.1167782-05:00.
- Middle Window:** Title: "script1.sql - DESKTOP-S6ILFL9\SQLEXPRESS.HRDB (DESKTOP-S6ILFL9\wang (62))". It contains scripts for creating a login AdminInterviewer, a user DrXie, and adding DrXie to the InterviewAdmin role. The output window shows "Commands completed successfully." and a completion time of 2022-12-11T17:03:51.1476054-05:00.
- Bottom Window:** Title: "script2.sql - DESKTOP-S6ILFL9\SQLEXPRESS.HRDB (DESKTOP-S6ILFL9\wang (66))". It contains scripts for creating a login AdminOnboarding, a user Andrew, and adding Andrew to the OnboardTeam and DocAdmin roles. The output window shows "Commands completed successfully." and a completion time of 2022-12-11T17:09:06.1501504-05:00.

# Appendix

script3.sql - DESKTOP-S6ILFL9\SQLEXPRESS.HRDB (DESKTOP-S6ILFL9\wang (57)) - Microsoft SQL Server Management Studio

```
/*script3: create Login AdminValidate, user IU */
CREATE LOGIN AdminValidate WITH PASSWORD = '987654',
    DEFAULT_DATABASE = HRDB;
GO

CREATE USER IU FOR LOGIN AdminValidate;
ALTER ROLE ComplaintTeam ADD MEMBER IU
```

110 % 110 %

Messages

Commands completed successfully.

Completion time: 2022-12-11T17:13:11.1399602-05:00

Query executed successfully.

script4.sql - DESKTOP-S6ILFL9\SQLEXPRESS.HRDB (DESKTOP-S6ILFL9\wang (60)) - Microsoft SQL Server Management Studio

```
/*script4: create Login DomainAdmin2, user dxie03*/
CREATE LOGIN DomainAdmin2 WITH PASSWORD = '030303',
    DEFAULT_DATABASE = HRDB;
GO

CREATE USER dxie03 FOR LOGIN DomainAdmin2;
ALTER ROLE ComplaintTeam ADD MEMBER dxie03
ALTER ROLE OnboardTeam ADD MEMBER dxie03
ALTER ROLE DocAdmin ADD MEMBER dxie03
ALTER ROLE InterviewAdmin ADD MEMBER dxie03
```

110 % 110 %

Messages

Commands completed successfully.

Completion time: 2022-12-11T17:17:14.0001167-05:00

Query executed successfully.

## 8. Screenshots for business reports

# Appendix

**create\_br1.sql**

```

USE HRDB;
GO

SELECT COUNT(*) AS [#applications for online job] FROM dxdb.Applications
WHERE dxdb.Applications.OpeningID IN
    (SELECT OpeningID FROM dxdb.JobOpenings JOIN dxdb.Jobs
     ON dxdb.JobOpenings.JobID = dxdb.Jobs.JobID
     WHERE Medium='Online')

SELECT COUNT(*) AS [#applications for onsite job] FROM dxdb.Applications
WHERE dxdb.Applications.OpeningID IN
    (SELECT OpeningID FROM dxdb.JobOpenings JOIN dxdb.Jobs
     ON dxdb.JobOpenings.JobID = dxdb.Jobs.JobID
     WHERE Medium='Onsite')

```

#applications for online job
5

#applications for onsite job
6

Query executed successfully.

**create\_br2.sql**

```

/*business report2: list all the average score who passed the whole interview.
Note that: a interview may contain many tests with same or different interviewer,
but by ID we can deduct which of them belongs to which whole interview(share the same time period)*/
USE HRDB;
GO

SELECT (TestId/10) AS InterviewBaseNum , AVG(Grade) AS [average score]
FROM dxdb.Tests
WHERE Result = 'pass'
GROUP BY (TestId/10)
ORDER BY (TestId/10) ASC

```

InterviewBaseNum	average score
401	100
402	93
405	99

Query executed successfully.

**create\_br3.sql**

```

/*business report3: find out is there a relationship
between candidates' response time and post platform
*/
USE HRDB;
GO

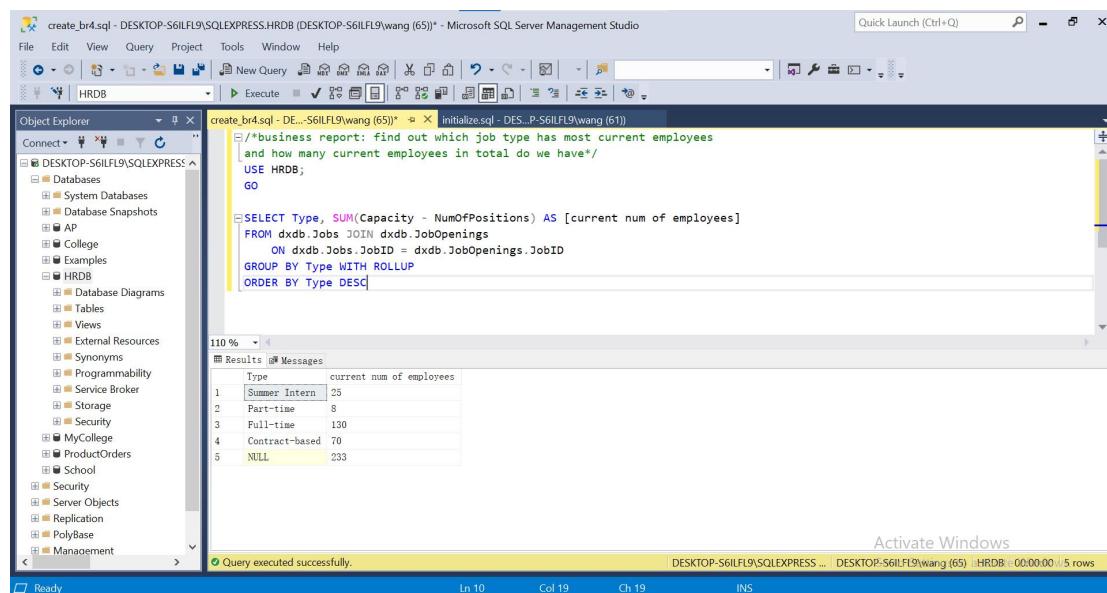
SELECT JobPlatform, AVG(DATEDIFF(day, PostDate, ApplyDate)) AS [AVGResponseTime (days)]
FROM dxdb.PlatformPosts JOIN dxdb.Applications
ON dxdb.PlatformPosts.OpeningID = dxdb.Applications.OpeningID
GROUP BY JobPlatform
ORDER BY AVG(DATEDIFF(day, PostDate, ApplyDate)) ASC

```

JobPlatform	AVGResponseTime (days)
Simplified Jobs	55
Company Mainpage	83
Indeed	85
LinkedIn	92
Handshake	96

Query executed successfully.

# Appendix



The screenshot shows the Microsoft SQL Server Management Studio interface. In the Object Explorer on the left, the database 'HRDB' is selected. The central pane displays a query window with the following SQL code:

```
create_br4.sql - DESKTOP-S6ILFL9\SQLEXPRESS.HRDB (DESKTOP-S6ILFL9\wang (65)) - Microsoft SQL Server Management Studio
File Edit View Query Project Tools Window Help
New Query Execute
-- initialize.sql - DESKTOP-S6ILFL9\wang (61)
/*business report: find out which job type has most current employees
and how many current employees in total do we have*/
USE HRDB;
GO

SELECT Type, SUM(Capacity - NumOfPositions) AS [current num of employees]
FROM dxdb.Jobs JOIN dxdb.JobOpenings
ON dxdb.Jobs.JobID = dxdb.JobOpenings.JobID
GROUP BY Type WITH ROLLUP
ORDER BY Type DESC;
```

The results pane shows a table with the following data:

Type	current num of employees
Summer Intern	25
Part-time	8
Full-time	130
Contract-based	70
NULL	233

At the bottom, a message indicates: "Query executed successfully." and "5 rows".