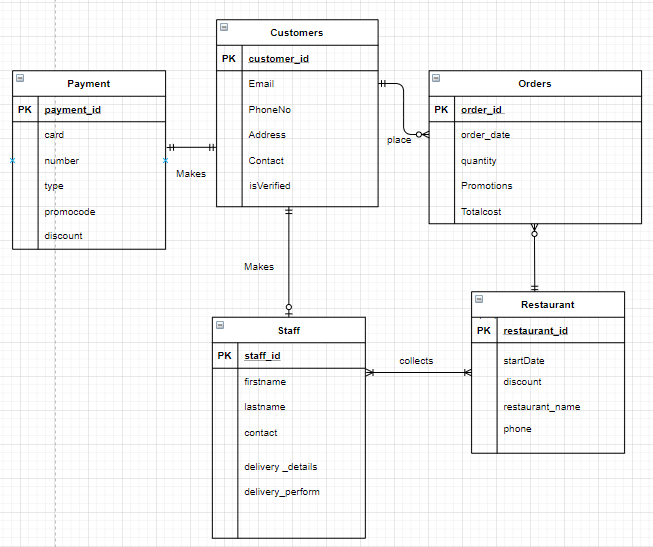
Q1a)

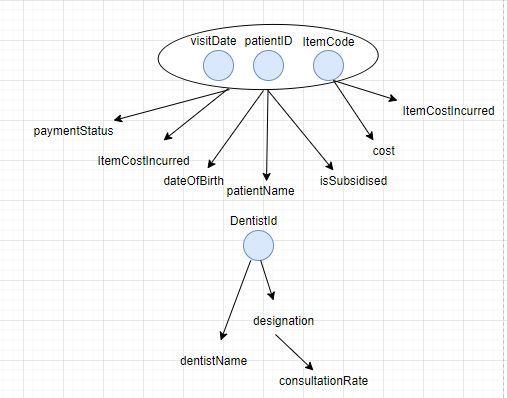


Q2a)

|  |
| --- |
| Functional Dependency |
| dentistId -> dentistName, designation  itemCode -> ItemCostIncurred, Cost  designation-> consultationRate  patientId, visitDate, ItemCode ->ItemCostIncurred, paymentStatus, isDiscountable  patientId -> patientName, dateOfBirth, isSusbsidised |

|  |
| --- |
| Multi Valued Dependency |
| designation ->-> dentistId  patientId ->-> ItemCode  patientId, visitDate ->-> ItemCode, ItemCostIncurred, paymentStatus, isDiscountable |

b)



c)

Remove the multivalued dependencies that are subsumed by the functional dependencies.

A multi valued dependency is subsumed by some functional dependency if all tuples for the multivalued dependency are in some relations for the functional dependencies.

|  |
| --- |
| designation ->-> dentistId  subsumed by  dentistId -> dentistName, designation |
| patientId, visitDate ->-> ItemCode, ItemCostIncurred, paymentStatus, isDiscountable  subsumed by  patientId, visitDate, ItemCode ->ItemCostIncurred, paymentStatus, isDiscountable |

Hence the multi valued dependency is absorbed by the functional dependency.

In this case, we will handle the functional dependencies first:

|  |
| --- |
| dentistId -> dentistName, designation  itemCode -> ItemCostIncurred, consultationRate, Cost  patientId, visitDate, ItemCode ->ItemCostIncurred, paymentStatus  patientId -> patientName, dateOfBirth, isSusbsidised |

***Database***

Visit (patientId, patientName, dateOfBirth, isSubsidised, dentistId, dentistName, designation, consultationRate, visitDate, itemCode, cost, isDiscountable, itemCostIncurred, paymentStatus)

1. Consider designation-> consultationRate

Visit2(patientId, patientName, dateOfBirth, isSubsidised, dentistId, dentistName, designation (FK), ~~consultationRate~~, visitDate, itemCode, cost, isDiscountable, itemCostIncurred, paymentStatus)

Move the attributes of dependency, that is the attributes that are determined into a new relation.

designation (designation (PK), ConsultationRate)

Make a copy of the determinant in the original relation and place the copy into the new relation. Check and see if that value exists in the multi-valued dependency, if it does exist, make that value the foreign key in the original table.

Create a referential Integrity constraint between the original relation and the new relation.

* Visit2.Designation must exist in designation.Designation

Every determinant in designation (designation (PK), ConsultationRate) is a functional dependency.

Repeat these steps for all functional dependency.

1. Consider dentistId -> dentistName, designation

Visit3(patientId, patientName, dateOfBirth, isSubsidised, dentistId, ~~dentistName~~, ~~designation~~, ~~consultationRate~~, visitDate, itemCode, cost, isDiscountable, itemCostIncurred, paymentStatus)

Dentist (dentistId(PK), dentistName, designation)

Create referential integrity constraint

* Visit3.dentistId must exist in Dentist.dentistId

1. Consider patientId -> patientName, dateOfBirth, isSusbsidised

Visit4(patientId (FK), ~~patientName~~, ~~dateOfBirth~~, ~~isSubsidised~~, dentistId, visitDate, itemCode, cost, isDiscountable, itemCostIncurred, paymentStatus)

Patient (patientId(PK) patientName, dateOfBirth, isSusbsidised)

Create referential integrity constraint

* Visit4.patient must exist in patient.patientId

Visit4. Designation must exist in designation. Designation

1. Consider itemCode -> ItemCostIncurred, Cost

* Visit5(patientId (FK), dentistId (FK), visitDate, itemCode(FK), ~~cost~~, isDiscountable, ~~itemCostIncurred~~, paymentStatus)

Item (itemCode(PK), ItemCostIncurred, Cost )

Create referential integrity constraint

Visit5.Item must exist in Item.Item

Visit6(patientId(FK), dentistId(FK), visitDate, itemCode(FK), isDiscountable, paymentStatus)

**Now we will look at the multi-valued dependency**

Consider designation ->-> dentistId

1. Visit6(patientId, ~~dentistId(FK),~~ visitDate, itemCode(FK), isDiscountable, paymentStatus)

Consider patientId ->-> ItemCode

Visit6(patientId(FK), visitDate, ~~itemCode(FK),~~ isDiscountable, paymentStatus)

Visit7(patientId(FK), visitDate, isDiscountable, paymentStatus)

**Final set of relation and referential integrity constraints:**

Dentist (DentistId(PK), dentistName, designation(FK)) BCNF

Item (itemCode(PK), ItemCostIncurred, Cost ) BCNF

Designation (Designation (PK)(FK), ConsultationRate) 4NF

Patient (patientId(PK)(FK) patientName, dateOfBirth, isSusbsidised) 4NF

Visit7 (PatientId(PK)(FK), visitDate, isDiscountable, paymentStatus ) 4NF

3a)i)

|  |
| --- |
|  |

3a)ii)

|  |
| --- |
| Create Function getAge(@dob date)  returns int  as  begin  declare @age int  set @age = CONVERT(int, DATEDIFF(DD, @dob,'2020/07/18')/365.25)  return @age  end |

3a)iii)

|  |
| --- |
| CREATE TABLE party  (  abbreviation VARCHAR(3) NOT NULL CONSTRAINT abbreviation PRIMARY KEY,  NAME VARCHAR(25) NOT NULL,  yearformed INT NOT NULL,  CHECK (yearformed >=1954 AND yearformed<=CONVERT(INT, Year(Getdate())))  )  CREATE TABLE division  (  divisionid VARCHAR(2) NOT NULL PRIMARY KEY,  CONSTRAINT divisionid CHECK(Len(divisionid)=2 AND divisionid NOT LIKE  '[[:<:]][^0-9][^aA-zZ][[:>:]]'),  seat SMALLINT NOT NULL,  voters BIGINT NOT NULL,  rejected INT NOT NULL,  CHECK(seat>=1 AND seat<=6 AND voters>=10000 AND rejected >0)  )  CREATE TABLE candidacy  (  candidacyid INT NOT NULL IDENTITY (1, 1) PRIMARY KEY,  divisionid VARCHAR(2) NOT NULL FOREIGN KEY REFERENCES division(  divisionid),  votesobtained INT NOT NULL,  samplevotesper100 INT NOT NULL  )  CREATE TABLE candidate  (  id VARCHAR(9) NOT NULL PRIMARY KEY,  NAME VARCHAR(30),  dateofbirth DATE,  party VARCHAR(3) FOREIGN KEY REFERENCES party(abbreviation),  candidacyid INT NOT NULL FOREIGN KEY REFERENCES candidacy(candidacyid),  CONSTRAINT id CHECK(Len(id)=9 AND id NOT LIKE '[^a-zA-Z]%')  ) |

Q3a)iv)

|  |
| --- |
| INSERT INTO party              (abbreviation,               NAME,               yearformed)  VALUES      ('JPP',               'Justice Progress Party',               1957)   INSERT INTO party              (abbreviation,               NAME,               yearformed)  VALUES      ('OPP',               'ONE PEOPLE PARTY',               1954)   INSERT INTO party              (abbreviation,               NAME,               yearformed)  VALUES      ('TWP',               'TOGETHER WE PARTY',               2020)   INSERT INTO division              (divisionid,               seat,               voters,               rejected)  VALUES      ('A1',               2,               81232,               549),              ('A2',               1,               31294,               491),              ('B1',               1,               29192,               325),              ('C1',               3,               129821,               738)   SET IDENTITY\_INSERT candidacy ON   INSERT INTO candidacy              (candidacyid,               divisionid,               votesobtained,               samplevotesper100)  VALUES      (1,               (SELECT divisionid                FROM   division                WHERE  voters = 81232),               52551,               65),              (2,               (SELECT divisionid                FROM   division                WHERE  voters = 81232),               27112,               35),              (3,               (SELECT divisionid                FROM   division                WHERE  voters = 31294),               12541,               40),              (4,               (SELECT divisionid                FROM   division                WHERE  voters = 31294),               18252,               60),              (5,               (SELECT divisionid                FROM   division                WHERE  voters = 29192),               14002,               49),              (6,               (SELECT divisionid                FROM   division                WHERE  voters = 29192),               4324,               14),              (7,               (SELECT divisionid                FROM   division                WHERE  voters = 29192),               9324,               37),              (8,               (SELECT divisionid                FROM   division                WHERE  voters = 129821),               59482,               45),              (9,               (SELECT divisionid                FROM   division                WHERE  voters = 129821),               69518,               55)   INSERT INTO candidate              (id,               NAME,               dateofbirth,               candidacyid,               party)  VALUES      ('A1111111A',               'Mohamad Faisal',               '1990-01-01',               1,               'OPP'),              ('A1111112A',               'Chia Hon Huat',               '1999-07-16',               1,               'OPP'),              ('A1111113A',               'Nurul Ahmad',               '1955-02-06',               2,               'TWP'),              ('A1111114A',               'Tan Chin Siong',               '1960-03-12',               2,               'TWP'),              ('A2222221A',               'Peter Thiagu',               '1970-04-17',               3,               'OPP'),              ('A2222222A',               'Ravi Pillay',               '1973-05-19',               4,               'JPP'),              ('B1111111B',               'Goh Hong Hui',               '1965-08-09',               5,               'OPP'),              ('B1111112B',               'Koh Li Choo',               '1969-09-23',               6,               'JPP'),              ('B1111113B',               'Ng Tiong Keng',               '1958-10-30',               7,               'TWP'),              ('C1111111C',               'Mohamed Azhar',               '1984-11-01',               8,               'OPP'),              ('C1111112C',               'Loganathan',               '1997-07-07',               8,               'OPP'),              ('C1111113C',               'Kelvin Ong',               '1995-04-17',               8,               'OPP'),              ('C1111114C',               'Vigneswan Menon',               '1992-03-26',               9,               'JPP'),              ('C1111115C',               'Lee Wei Zhong',               '1994-02-04',               9,               'JPP'),              ('C1111117C',               'Abiram Raj',               '1990-01-19',               9,               'JPP') |

Q3b)i)

|  |
| --- |
| SELECT \*,dbo.getAge(dateOfBirth)as age from Candidate order by age DESC, party ASC |

Q3b)ii)

|  |
| --- |
| SELECT \* from Division  where divisionId in  (SELECT divisionId from candidacy  where candidacyID in  (SELECT candidacyID  from Candidate  where dbo.getAge(dateOfBirth)<25)) |

Q3b)iii)

|  |
| --- |
| SELECT Division.divisionId,division.seat ,Candidate.party  from Division  inner join Candidacy  ON Division.divisionId = Candidacy.divisionId  inner join Candidate  ON Candidacy.candidacyId = Candidate.candidacyID  AND Candidacy.votesObtained=(SELECT MAX(votesObtained) from Candidacy where Division.divisionId=Candidacy.divisionId)  group by Division.divisionId,division.seat ,Candidate.party, Candidacy.votesObtained |

Q3b)iv)

|  |
| --- |
| SELECT party.abbreviation, name, ISNULL(sum(seat),0) as "Total seats won" from WinnersofDivision  full outer join party on WinnersofDivision.party = party.abbreviation group by abbreviation, name |

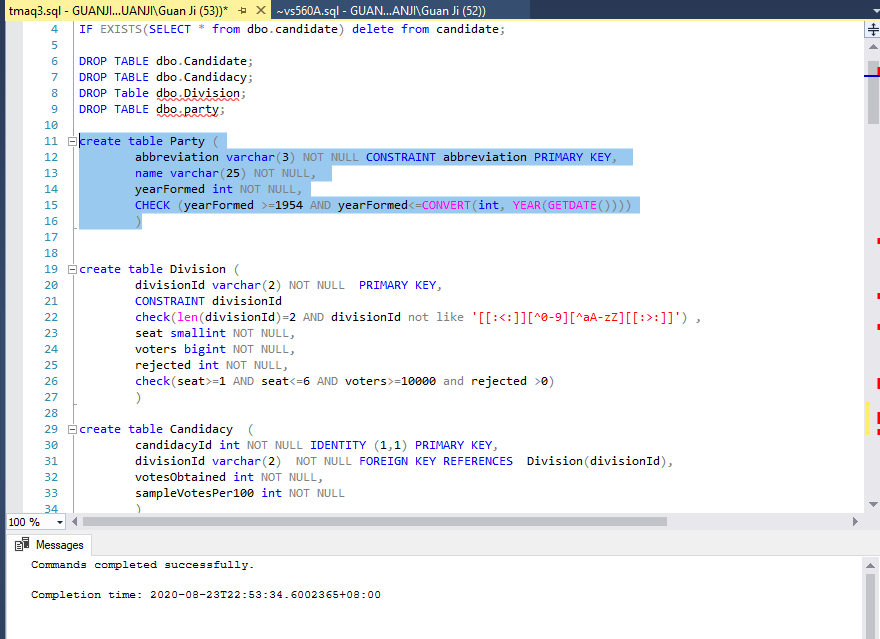
Q3c)i)

|  |
| --- |
|  |

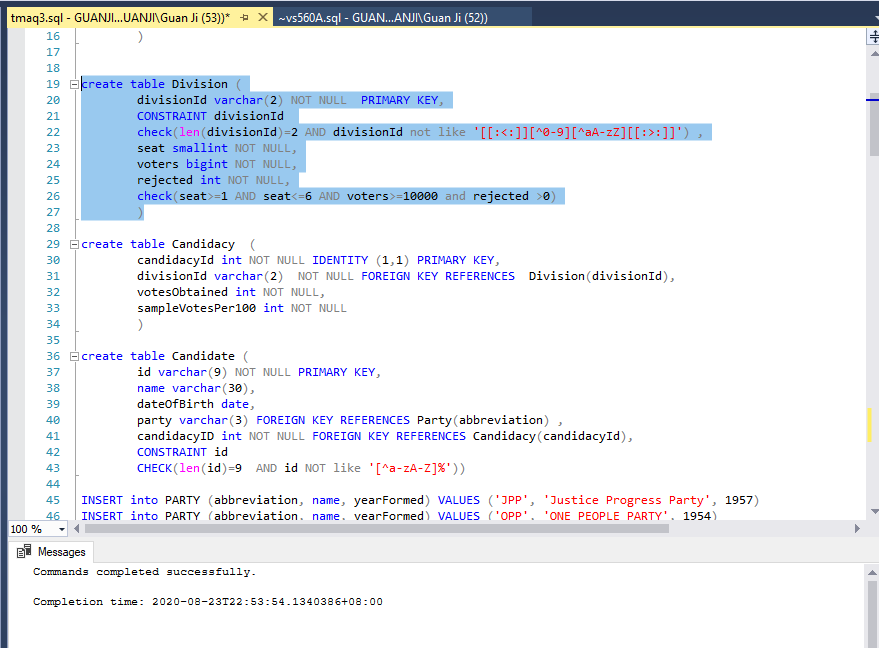
Q3c)ii)

|  |
| --- |
|  |

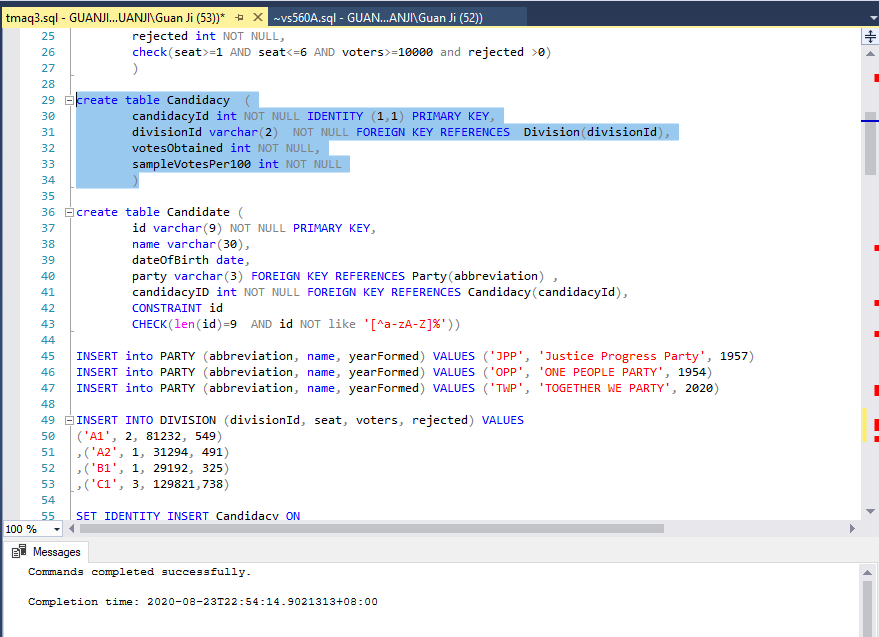
Create party table



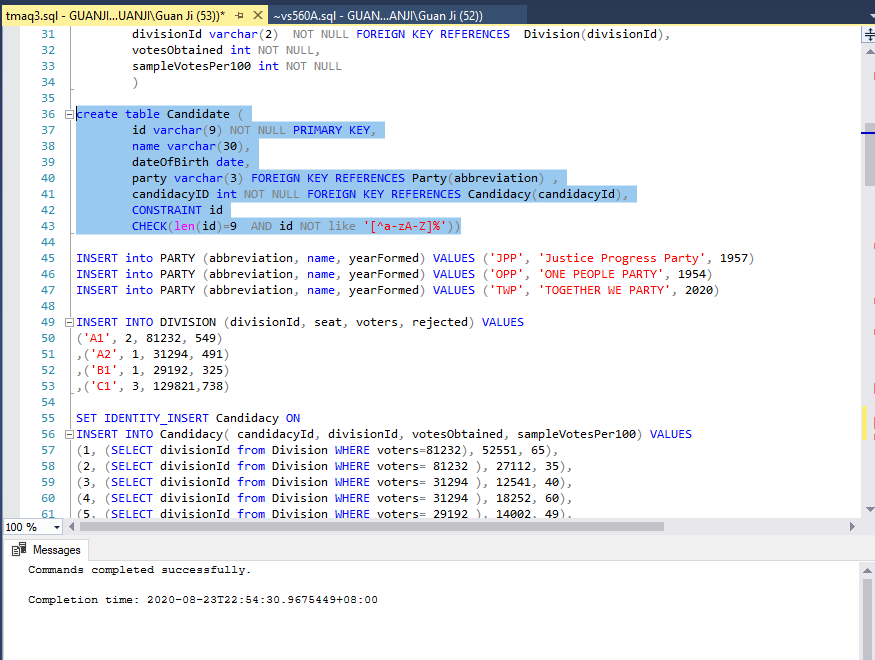
Create division table



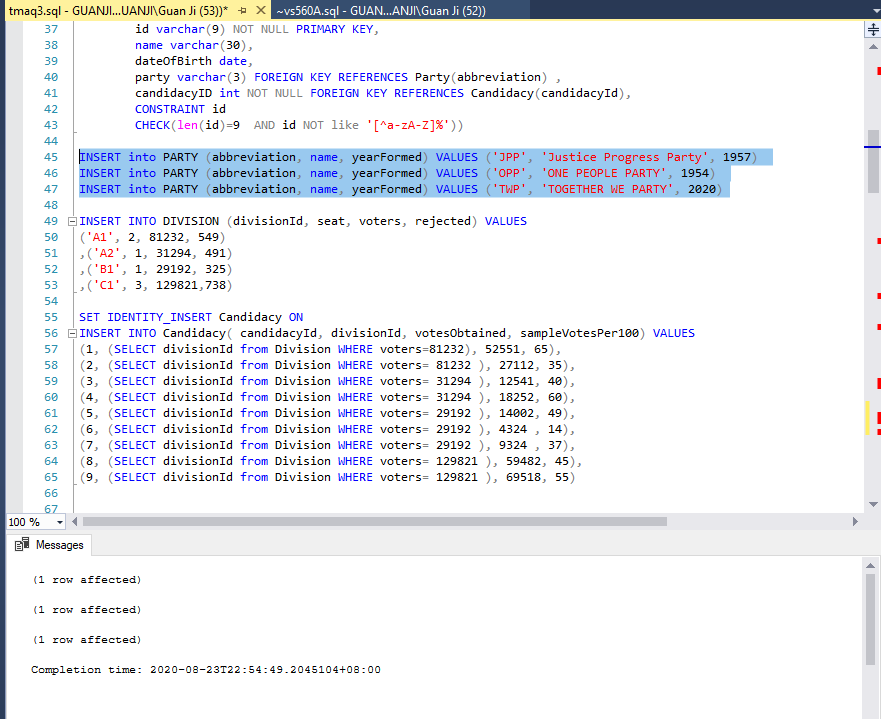
Create candidacy table



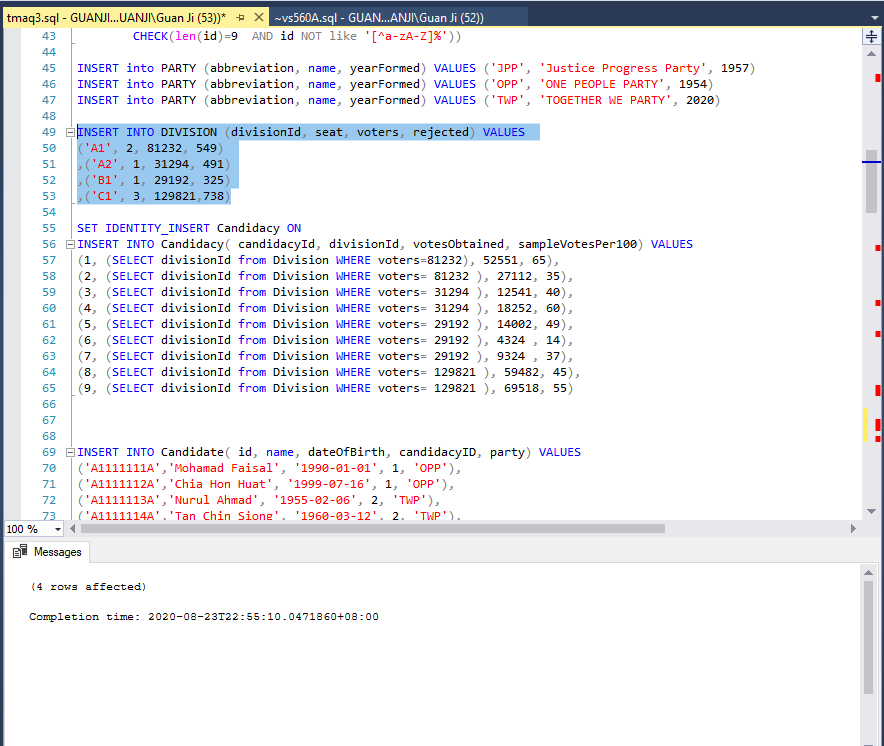
Create candidate table



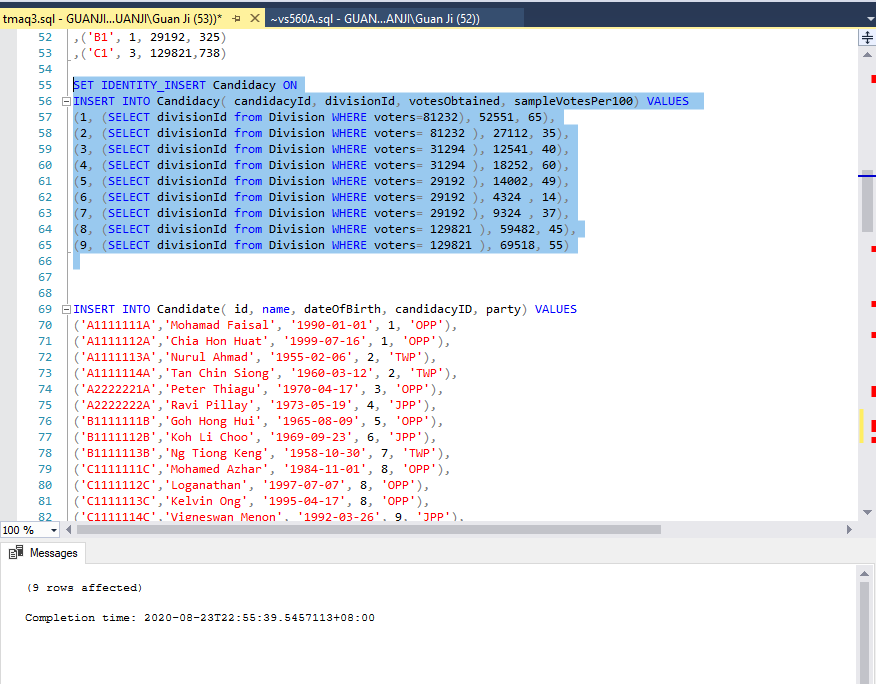
Insert into party table



Insert into division table



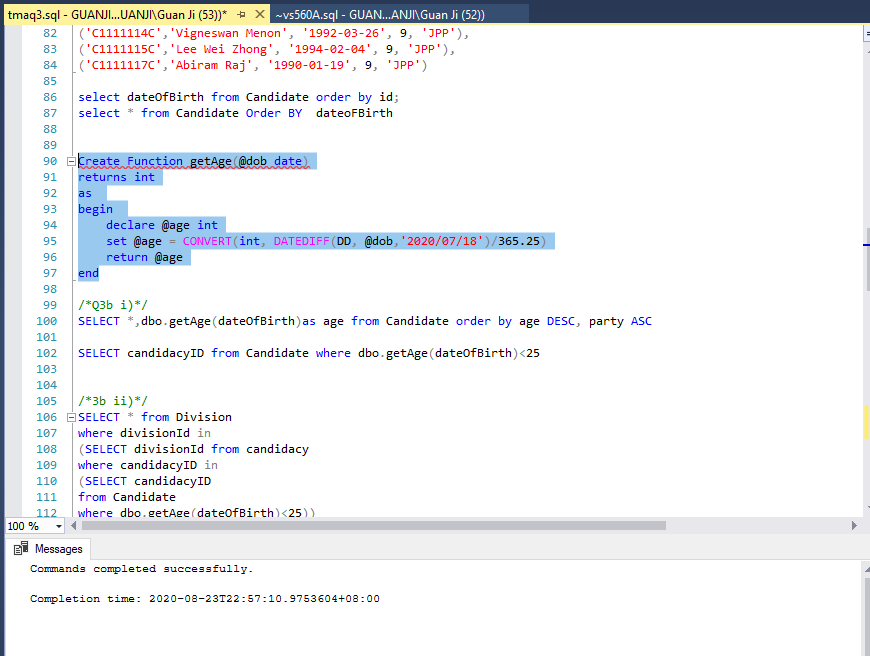
Insert into candidacy table

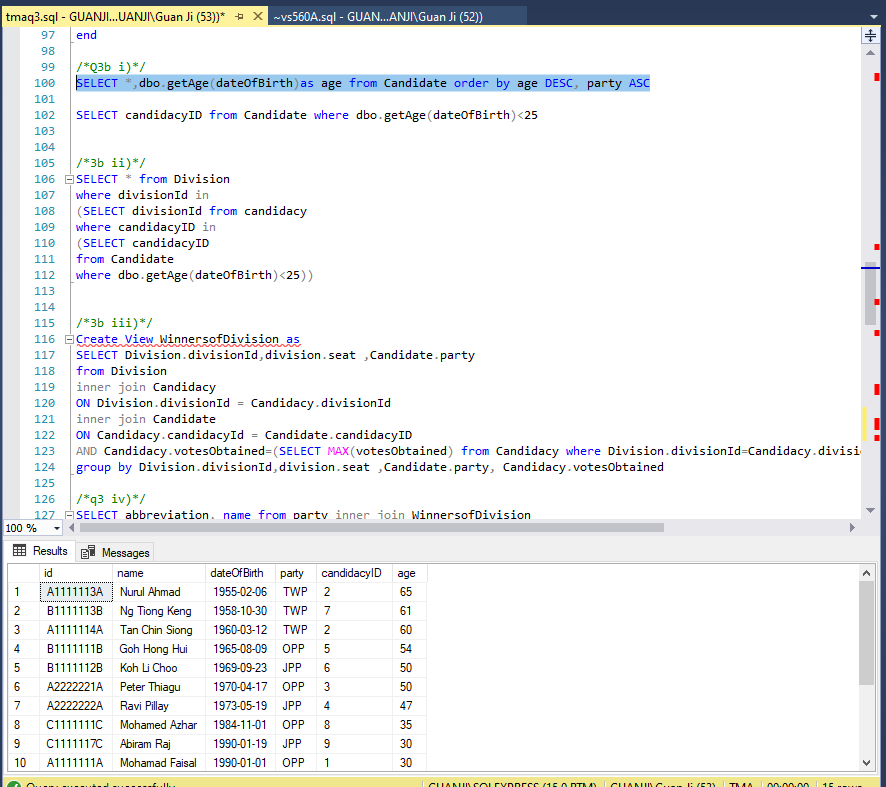


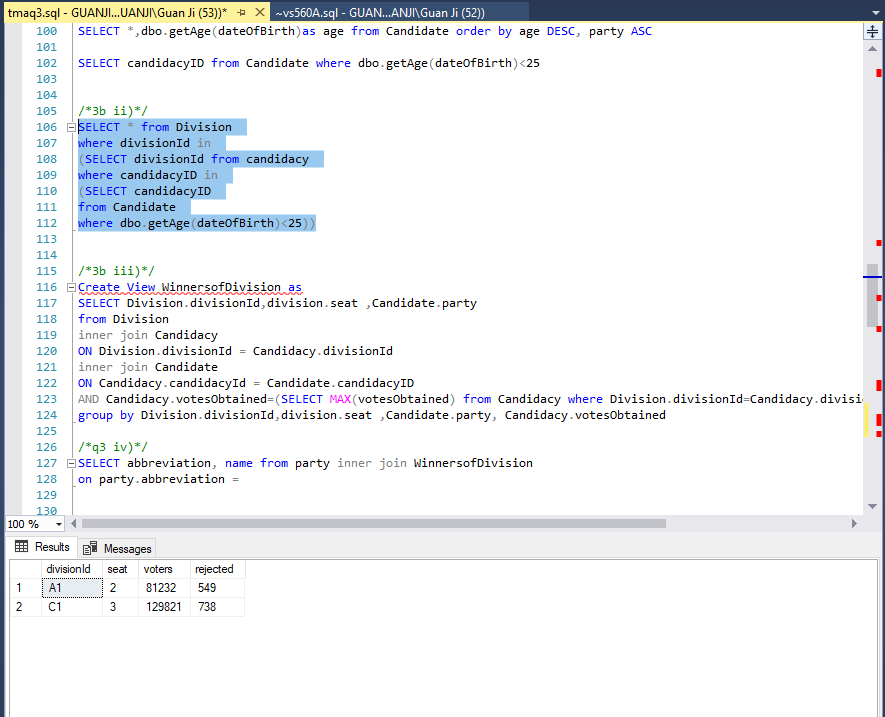
Insert into candidate table

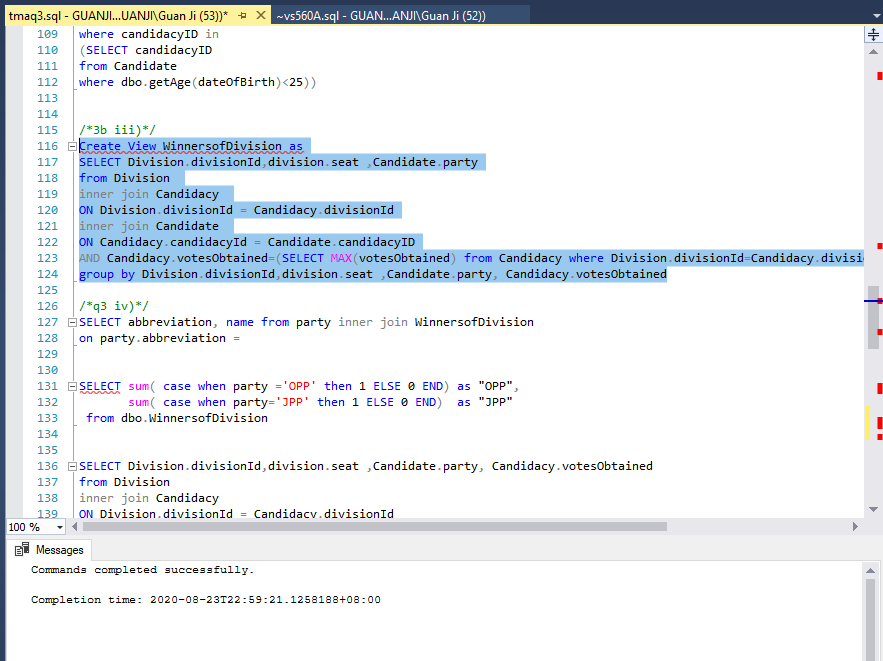


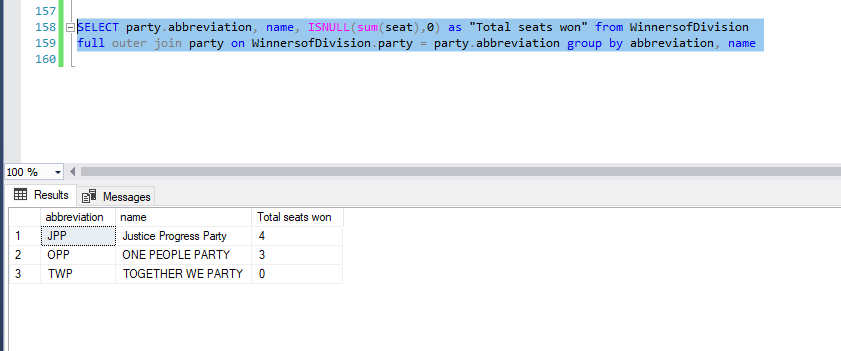
Create function to get age











4a)

When 2 concurrent transactions try to read and update the same data, lost update may take place, the system administrator should practice isolation, no other transaction should take place.

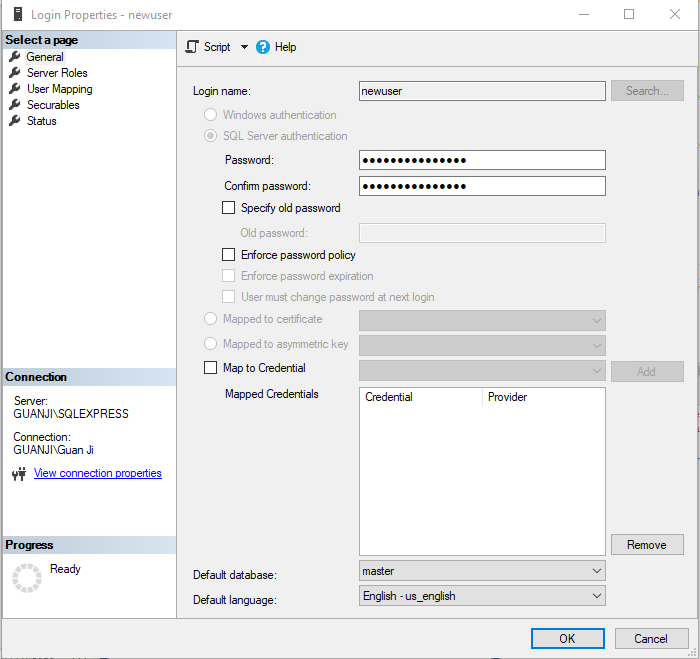
Lost update will not occur because only one patientId is making a dental appointment unless, the system allows for multiple updates. We will now consider that there is only one patient that can be entered into the system at once.

A Deadlock will be more likely to occur in the Update statement. There is a lock on dentist, and it needs a lock on appoint to finish the transaction. Insert statement has a lock on appoint and needs a lock on dentist to finish the transaction. So, both transactions will not be complete because both are lock by each other. We will need to cancel this if the deadlock occurs.

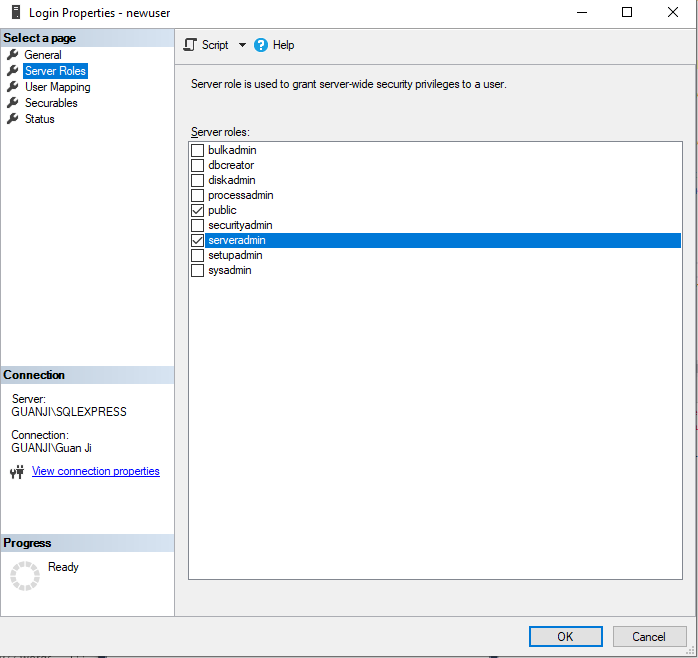
Q4b)

The two ways are authorisation and server roles

Database Administrator can create more users in the database and allow only specific users to execute query command in the database. By logging in, the credentials are then authentication and authorisation, to verify if a username and password matches the one that was created by the DBA and only allow the specific user to run the query for the schema.



In this diagram above, I am creating testUser and only assigning the schema TMA to the newuser



Additionally, I can grant him server explicit roles to this user based on what I want to provide to him.

Q4c)

|  |  |  |
| --- | --- | --- |
| Transaction | Type of Operation | Description |
| T1 | Start | Start of the Operation |
| T1 | Insert | Inserting the query statement into the database dentist table at rowed s12.. |
| T2 | Start | Start of the next operation |
| T2 | Update | Update query is executed into appointment table from null to row 25 onwards, p02.. |
| T2 | Insert | Insert into reminder table rowID 132 new insertion, 132, p02,… |
| T1 | Commit | Saving of the record in the database |
| T3 | Start | Start of the Operation |
| T3 | Update | Update appointment table of rowID 18, p15 |
| T4 | Start | Start of the Operation |
| T4 | Delete | Remove rowID 19 of Appointment table |
| There might be some issues with the delete statement at rowID, ma be a foreign key issue hence delete statement is unable to execute. | | |
| T3 | Rollback | Oh no.. a query execution failed... rollback every statement.  Rollback before crash, apply before image and rollback do not restart |
| T4 | Commit | Saving of the record in the database |

Atomicity take place here, the entire transaction takes place at once or none of the query takes place at all, hence we roll back. Isolation is being practice here, one transaction is executed at a time, update then insert.

The database recovery procedure from T1 to T4

The reverse pointer of the first database operation is 0, as there is no database operation preceding the first database operation. The forward pointer of the last database operation is 0, as there is no database operation following the last database operation.