School of Engineering and Computer Science

SWEN 304 **Database System Engineering**

Assignment 1 - 300492683

The objective of this assignment is to test your understanding of database foundations, basic terms, and the relational data model the entity relational model. It is worth 15% of your final grade. The assignment is marked out of 100.

The assignment is due on **Friday, 26 March, 23:59 pm**. Please submit your assignment in **pdf** via the submission system.

Question 1 [5 marks]

a) **[2 marks]** Give your own example of a relation schema with at least three attributes to illustrate what a relation schema is.

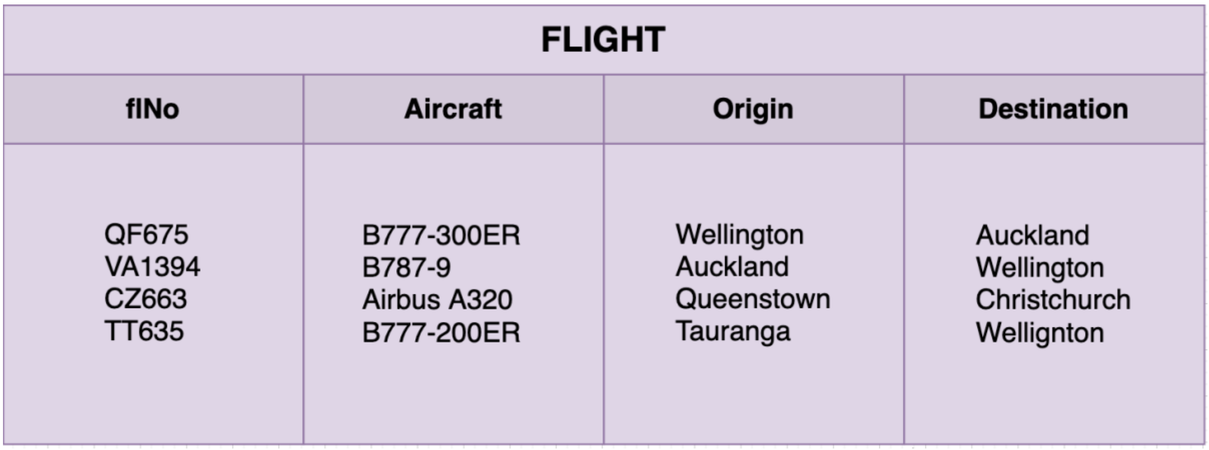
Answer:

Flight(flNo, Aircraft, Origin, Destination)

b) **[3 marks]** Give a relation (represented as a table) with at least four tuples over your relation schema in part **a)**. Explain how your relation satisfies the properties of schema keys.

Satisfies as no duplicate keys.

Answer:



FLIGHT

flNo	Aircraft	Origin	Destination
QF675	B777-300ER	Wellington	Auckland
VA1394	B787-9	Auckland	Wellington
CZ663	Airbus A320	Queenstown	Christchurch
TT635	B777-200ER	Tauranga	Wellignton

Question 2 [15 marks]

Suppose you are the manager of an IT company in Wellington, and you are using a relational database to manage your business data. The following table shows an instance of the STAFF relation schema that stores basic data on the employees working in your company.

STAFF

Name	DoB	EmpNo	JobTitle
Andy	22/01/1988	88-11	Developer
Mickey	11/02/1996	96-02	Project Manager
Jerry	22/02/1989	89-08	Developer
Andy	15/05/1990	90-01	Business Analyst
Alice	22/02/1989	89-06	Developer
Mary	12/07/1990	90-04	Architect
Mary	25/11/1996	96-22	Tester

a) **[8 marks]** For every set of attributes (that is, for every subset of the set {Name, DoB, EmpNo, JobTitle}) decide whether you can deduce that it is *not* a candidate key, assuming the instance is legal. Justify your answer.

Answer: Name, DoB and JobTitle are not candidate keys. DoB and EmpNo are candidate key as they have no repeated attributes.

b) **[4 marks]** For every remaining set of attributes (that is, every set not ruled out as a candidate key in part **a)**), discuss whether you consider it a suitable candidate key? Justify your answer.

Answer: EmpNo is a suitable candidate key as every employee number should differ and meets all the criteria as a candidate key, as well as all employees should have an employee number as they begin. Logically DoB could have duplicates.

c) **[2 marks]** Which of the candidate keys identified in part **b)** would you choose as the primary key?
Answer: EmpNo would be chosen as the primary key because there will not be any duplicates as an employee ID is always unique.

d) **[1 mark]** Add a tuple with your own data into the STAFF relation. How would you check that the primary key identified in part **c)** is still valid?

Answer: This tuple should not affect the primary key identified as the tuple contains unique values that are not the same as any other in the table.

Andree	06/10/2000	98-16	Business Analyst

Question 3 [10 marks]

Suppose your software company has developed a relational database for the grocery store “Fruits and more”. The underlying database schema contains the following relation schemas:

- COMPANY (Cid: STRING, Name: STRING, Location: STRING) with primary key {Cid}
- FRUITS (Fid: STRING, Name: STRING, Cid: STRING, InStock: INTEGER, Price: INTEGER) with primary key {Fid, Cid} and foreign key Cid $\subseteq$ COMPANY[Cid]

Below you find instances of these two relation schemas:

FRUITS

Fid	Name	Cid	InStock	Price
557	Apple	23XY	50	21
85520	Pear	A15F	0	78
63311	Pear	FVT35	211	49
36773	Kiwi	23XY	50	21
36773	Kiwi	FVT35	29	22

COMPANY

Cid	Name	Location
23XY	GreatFruits	Wellington
FVT35	Yummy	Wellington
F15A	GreatFruits	Levin
A15F	BetterFruits	Lower Hutt
5AB32	NiceFruits	*null*

Your tasks are as follows.

a) **[5 marks]** Decide which of the following tuples can be added or removed, respectively. *Justify your answers!*

1. Insert tuple (‘XYZ4’, ‘Wellington’, ‘Yummy’) into COMPANY

Answer: YES the tuple can be added, because the CID does not exist in the company relationship.

2. Insert tuple (*null*, ‘Tasty’, ‘Wellington’) into COMPANY

Answer: NO the tuple cannot be added, because the primary key value (Cid) must not be null.

3. Insert tuple (‘FVT35’, ‘SweetFruits’, ‘Porirua’) into COMPANY

Answer: NO the tuple cannot be added, because multiple primary key’s that are of the same value cannot exist.

4. Delete tuple (‘A15F’, ‘BetterFruits’, *null*) from COMPANY

Answer: NO the tuple cannot be deleted, because the location specified is incorrect and cannot be deleted.

5. Delete tuple (‘23XY’, ‘GreatFruits’, ‘Wellington’) from COMPANY

Answer: NO the tuple can and cannot be deleted, however if you delete the whole tuple in fruit it can be deleted.

b) **[5 marks]** Decide which of the following tuples can be added or removed, respectively. *Justify your answers!*

1. Insert tuple (‘55555’, *null*, ‘F15A’, 2, 99) into FRUITS

Answer: YES the tuple can be added, because name cannot be null.

2. Insert tuple (‘54556’, ‘Lemon’, ‘FV35’, 20, 43) into FRUITS

Answer: NO the tuple cannot be added, because the cid does not exist in COMPANY table therefore there is no relationship.

3. Insert tuple (‘53557’, ‘Apple’, ‘5AB32’, 500, 1) into FRUITS

Answer: YES the tuple can be added, because everything mentioned is correct and the Cid matches.

4. Delete tuple (‘36773’, ‘Kiwi’, ‘23XY’, 50, 21) from FRUITS

Answer: Yes the tuple can be deleted, because the Fid and Cid matches up.

5. Delete tuple (‘46557’, ‘Apple’, ‘23XY’, 1, 21) from FRUITS

Answer: NO the tuple cannot be deleted, because the Fid does not exist and cannot be deleted.

Question 4 [25 marks]

Suppose your software company is planning to build a relational database for a new event booking system. The following relation schemas are part of the underlying database schema.

- EVENT (eventId, venue)
- CLIENT (emailAddress, name, dob, phone) with primary key {emailAddress}
- MANAGER (managerId, name) with primary key {managerId}
- BOOKING (managerId, eventId, emailAddress, date, noOfTickets, promoCode) with primary key {eventId, emailAddress, date}

The following additional constraints are known:

1. Each client may only use a single emailAddress.
2. Managers may also be clients but may not book an event for themselves.
3. For each event booking, the noOfTickets must be specified, while promoCode may be left blank (if not available).
4. An event may have up to four venues.

Your tasks are as follows:

a) [3 marks] For the relation schema EVENT, identify all suitable candidate keys. Explain your answer.

Answer: {eventId,venue} Is the only suitable candidate key as these must be unique tuples and are unlikely to be null.

b) [5 marks] For each of the relation schemas, identify all suitable foreign keys (if there are any). Explain your answer.

Answer:

MANAGER, CLIENT, EVENT does not have a foreign key. This is because there are no attributes in this relation schema used in other relation schemas as primary key.

BOOKING:
{emailAddress} connects CLIENT and EVENT because it is a primary key for the client and email address is an attribute for event.

{managerID} connects MANAGER and EVENT because it is a primary key for the cient and manager ID is an attribute for event.

c) [3 marks] Is it possible to add an event booking to the database with the emailAddress of a client who is not listed in the CLIENT relation? Explain your answer.

Answer: No, because the emailaddress is a foreign key for the relation schema BOOKING and a primary key for CLIENT. It will violate referential integrity constraints as adding a tuple with a foreign key with the paired primary does not exist. You cannot add the email address of a client who does not exist in the schema. In other words, the email address is a primary key in the CLIENT relation and if there is no email with the booking there is no relationship between BOOKING and CLIENT.

d) **[4 marks]** If the attribute eventId would not be part of the primary key of BOOKING, what would be the consequence? Explain your answer.
Answer: These consequences will allow null values and duplicates, there will be multiple events occurring in the same venue. Customers will not be able to purchase multiple tickets at the same venue. There will be a violation of uniqueness constraints if there are tuples with the same emailAddress and date.

e) **[5 marks]** Suppose, a client (‘paula@vuw.ac.nz’, ‘Paula’, 22/01/2000, ‘381-1230’) in the CLIENT relation has made several bookings stored in the relation BOOKING. When deleting the record of this client from the CLIENT relation, all her bookings should be deleted, too. How would you ensure this requirement? Explain your answer.

Answer: To ensure this requirement, ON DELETE CASCADE is the answer. The attributes for the foreign key {emailAddress} in relation BOOKING should be referenced to the primary key in relation CLIENT and then add the constraint of ON DELETE CASCADE. This means that whenever a record of a client is deleted it will automatically delete all records with the same foreign/primary key in BOOKING. So, the client Paula will have all her records deleted.

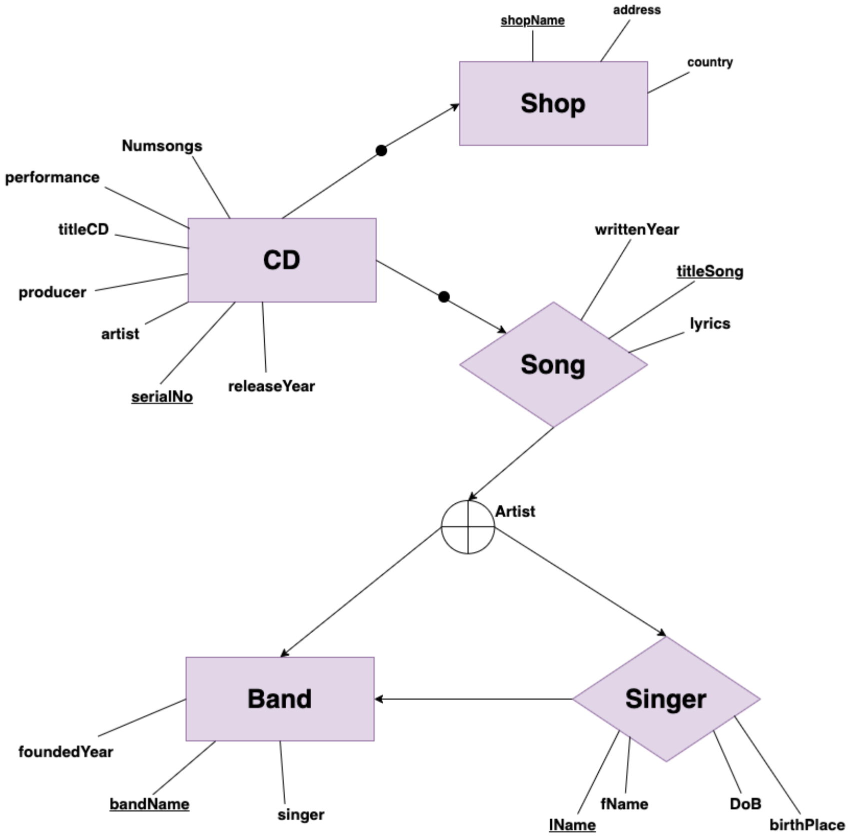
f) **[5 marks]** Suppose, a person (555, ‘Mary’) in the MANAGER relation is no longer working for the event booking system. When deleting the record of this manager from the MANAGER relation, all the bookings she made for clients should not be lost. How would you ensure this requirement? Explain your answer.

Answer: To ensure this requirement, DELETE SET NULL is the answer. This is because whenever a record of a manager is deleted, it will automatically set the foreign key to null to all the records with the same foreign/primary key. This means that the Manager Mary will have their records set as null.

Question 5 [30 marks]

You are asked to design a CD_COLLECTION database for your grandma's CD collection. A CD has a title, a release year, a unique serial number and was produced by a certain producer. A CD contains performances of songs by an artist. There are at most 20 songs on a CD. A song has a title, the year in which it was written, and the song lyrics. An artist is a singer or a band. A singer has a name (first name and last name), a birthday and a country of birth. A band has a band name and the year in which it was founded. A singer can be a member of a band. Furthermore, your grandma buys CDs at certain shops, which have a name, an address, and a country.

a) [24 marks] Draw an extended ER diagram for the database above. Write down the corresponding extended ER schema, including declarations of all the entity types (showing attributes and keys) and relationship types (showing components, attributes and keys).



Level 0: CD(titleCD, releaseYear, serialNo, producer, Numsongs, artist, performance)
Level 0: Shop(shopName, address, country)
Level 1: Song(titleSong, writtenYear, lyrics)
Level 2: Artist(singer, band)
Level 3: Band(bandName, foundedYear, singer)
Level 4: Singer(FName, LName, DoB, birthPlace)

Justification

Song is a relationship because the entities Band and Singer, which is related to the cluster Artist, share information from Song. Singer is a relationship because it has entities Band which is shared information. Artist is declared as a cluster because it can be both Singer and Band under the same property, and artist cannot be an entity or relationship as it does not contain any attributes but connections to Singer and Band. CD, Band and Shop are entities as they store information about of their objects and are not relationships because they are not necessarily used to share information for the rest of the database.

b) [6 marks] There may be information, requirements or integrity constraints that you are not able to represent in your diagram. Give three examples of integrity constraints that have not been represented in your diagram.

Remark: Whenever you feel that information is missing in the problem description above, add an assumption and make your assumption explicit. In practice you would consult the domain experts or potential users for clarification.

Answer:

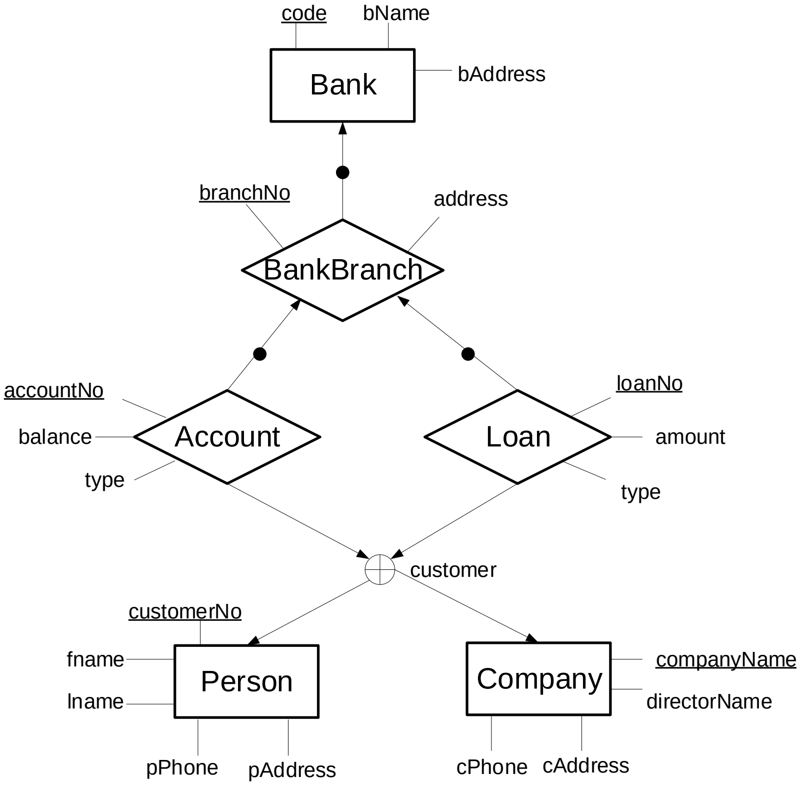
1. Foreign Key constraints: these constraints should typically state that for every x in the xTable should also be a member of the yTable. For example, for every song in the CD table, song must be a member of the Song table.

2. Not Null constraints: these constraints should be assigned to the most important components of each object in the database. For example, serialNo in CD should never be assigned null.

3. Unique constraints: these constraints define that an attribute must be unique, not null and prevents duplicates from existing in the table. For example, serialNo in CD must always be unique in the table and it should never contain duplicates.

Question 6 [15 marks]

Consider the extended ER diagram below.



a) [5 marks] Present the extended ER schema of the extended ER diagram above.

Bank(<u>code</u>, bName, bAddress) – level 2
BankBranch(<u>branchNo</u>, address) – level 3
Account(<u>accountNo</u>, balance, type) – level 2
Loan(<u>loanNo</u>, amount, type) – level 2
customer (cluster) – level 1
Person(<u>customerNo</u>, fname, lname, pPhone, pAddress) – level 0
Company(<u>companyName</u>, directorName, cPhone, cAddress) – level 0

b) [10 marks] Transform your extended ER schema into a relational database schema. In particular, list all the relation schemas in your relational database schema. For each relation schema, list all attributes, the primary key, the NOT NULL constraints, and the foreign keys.

Answer:

--- Bank (<u>code</u>, bName, bAddress) with primary key {code}, Not Null {code, bName, bAddress} and foreign keys {}

--- BankBranch (<u>branchNo</u>, address) with primary key {branchNo}, Not Null {branchNo} and foreign keys {}

---Account (<u>accountNo</u>, balance, type) with primary key {accountNo}, Not Null {accountNo, balance} and foreign key type $\subseteq$ Loan[type]

---Loan (loanNo, amount, type) with primary key {loanNo}, Not Null {loanNo, type} and foreign key foreign key type ⊆ Loan[type]

---Person (customerNo, fName, lName, pPhone, pAddress) with primary key {customerNo}, Not Null {customerNo, fName, lName} and foreign key

---Company (companyName, directorName, cPhone, cAddress) with primary key {companyName}, Not Null {companyName. directorName, cPhone} and foreign key

****