**Question #01: Identify all possible Candidate Keys (except primary keys) in the given schema, stating any assumptions you make.**

1. Branch (branchNo, street, city, postcode)

We might can say that street and postcode is unique in this table, but it can’t be selected as a candidate key because there can be multiple branches in the same street having similar postcode. So no candidate key other than primary key.

1. Staff (staffNo, fName, IName, position, sex, DOB, salary, branchNo)

There is no candidate key other than the primary key because every other value can be duplicated.

1. PropertyForRent (propertyNo, street, city, postcode, type, rooms, rent, ownerNo, staffNo, branchNo)

Even if we make propertyNo, and owerNo a composite key, then we cannot consider this composite key as a candidate key because it will make it a super key not key. Thus, no candidate key can be made other than the primary key.

1. Client (clientNo, fName, IName, telNo, prefType, maxRent, eMail)

eMail can be considered as a candidate key if we use unique keyword along with it.

1. PrivateOwner (ownerNo, fName, IName, address, telNo, eMail, password)

Email can be made a candidate key if we use unique keyword along with it

1. Viewing (clientNo, propertyNo, viewDate, comment)

No candidate key other than the primary key.

1. Registration (clientNo, branchNo, staffNo, dateJoined)

No candidate key other than the primary key.

**Question #02: Identify all the FOREIGN KEYS present in the given schema, stating any assumptions**

**you make.**

* Staff and Branch will have branchno as foreign key. Branch will be base table
* PropertyForRent and Staff will have StaffNo as foreign key. Staff will be the referenced table.
* PropertyForRent and Branch will share branchno as foreign key. Branch will be base table
* Viewing and PropertyForRent shares propertyNo. PropertyForRent will be base table.
* Viewing and client will share clientNo as foreign key. Client will be the base table.
* Client and Registration will share clientNo as foreign key. Client will be the base table.
* Staff and Registration will share staffNo. Staff will be the base table.

**Question #03: Suppose that each of the following Update operations is applied directly to the database state shown in figure 1, discuss all constraints violated by each operation, if any?**

* Insert <SA9, 'Scott', 'Jeff', 'Clerk', 1, '1977-10-01', 58000, B007> into STAFF

Primary key constant is violated, as duplicates can’t be entered in the primary key. Domain constraint is also violated because sex is a char not a integer data type.

* Insert <CR56, '', '', '1990-12-07'> into REGISTRATION

Since primary key is the composite key of clientNo and branchNo, both the columns can’t be kept null. However in this case branchNo is null which violates the primary key constraint.

* Delete the Branch tuple with branchNo = ‘B002’

No constraint is violated as B002 is not used in any of the table which has the foreign key of branchNo.

* Delete the PrivateOwner tuple with the Name = ‘Tony Shaw’

Referential Integrity constraint is violated, because the ownerNo is foreign key in PropertyForRent which uses the ownerNo of Tony Shaw CO93. If we’re considering Name as a separate column, than the column name doesn’t exists as well.

* Modify the clientNo of the VIEWING tuple with clientNo = ‘CR62’ to ‘CR97’

Referential Integrity constraint is violated, because the clientNo ‘CR97’ doesn’t exist in the base table’s primary key column.

* Modify the propertyNo attribute of the PropertyForRent tuple with propertyNo = ‘PA14’ to ‘PA16’.

It will violate referential integrity constraint as the Viewing table is using propertyNo ‘PA14’.

* Modify the branchNo attribute of the PropertyForRent tuple with branchNo = ‘B007’ to NULL

No constraint is violated, as the foreign key can be kept null in the referencing table.

**Question #04: Write appropriate SQL DDL statements to define each of the following tables in the database with the specified additional constraints. The defined statements should execute the constraint efficiently when updates are applied to the database.**

**Staff**

No staff member should be older than 60 years as of today’s date

Salary must remain in between 9000 to 30000.

Every attribute of staff must have some valid value

Note: Branch table is not asked in question but for branchNo in Staff to make it a foreign key, we’ve to first make a table Branch and make branchNo primary key in it.

CREATE TABLE Branch(

branchNo VARCHAR2(5) PRIMARY KEY,

street VARCHAR2(20),

city VARCHAR2(10),

postcode VARCHAR2(10)

);

CREATE TABLE Staff(

staffNo VARCHAR2(10) PRIMARY KEY,

fName VARCHAR2(10) NOT NULL,

IName VARCHAR2(10) NOT NULL,

position VARCHAR2(10) NOT NULL,

sex CHAR NOT NULL,

DOB DATE NOT NULL CHECK(TO\_CHAR(DOB,'DD-MON-YYYY') > '09-OCT-1961'),

salary NUMBER(6) NOT NULL CHECK (salary >= 9000 and salary <= 30000),

branchNo VARCHAR2(5) NOT NULL,

FOREIGN KEY (branchNo) REFERENCES Branch(branchNo)

);

**PropertyForRent**

‘House’ should be the default type for any property.

A property must at least have 3 rooms

Note: Staff and PrivateOwner table is not asked in question but for staffNo and ownerNo in PropertyForRent, we’ve to make it a foreign key for which we’ve to first make a table Staff and PrivateOwner and make staffNo, ownerNo primary key in it, but since we’ve to make PrivateOwner in next part, I won’t re-write in this part.

CREATE TABLE Staff(

staffNo VARCHAR2(10) PRIMARY KEY,

fName VARCHAR2(10) NOT NULL,

IName VARCHAR2(10) NOT NULL,

position VARCHAR2(10) NOT NULL,

sex CHAR NOT NULL,

DOB DATE NOT NULL CHECK(TO\_CHAR(DOB,'DD-MON-YYYY') > '09-OCT-1961'),

salary NUMBER(6) NOT NULL CHECK (salary >= 9000 and salary <= 30000),

branchNo VARCHAR2(5) NOT NULL,

FOREIGN KEY (branchNo) REFERENCES Branch(branchNo)

);

CREATE TABLE PropertyForRent (

propertyNo VARCHAR2(10) PRIMARY KEY,

street VARCHAR2(20),

city VARCHAR2(10),

postcode VARCHAR2(10),

type VARCHAR2(10) DEFAULT 'House',

rooms NUMBER(2) CHECK(rooms>=3),

rent NUMBER(5),

ownerNo VARCHAR2(10),

staffNo VARCHAR2(10),

branchNo VARCHAR2(5),

FOREIGN KEY (ownerNo) REFERENCES PrivateOwner(ownerNo),

FOREIGN KEY (staffNo) REFERENCES Staff(staffNo),

FOREIGN KEY (branchNo) REFERENCES Branch(branchNo)

);

**PrivateOwner**

telNo must be DISTINCT for every Owner

eMail must be DISTINCT for evert Owner

CREATE TABLE PrivateOwner(

ownerNo VARCHAR2(10) primary key,

fName VARCHAR2(10),

IName VARCHAR2(10),

address VARCHAR2(10),

telNo VARCHAR2(20) UNIQUE,

eMail VARCHAR2(20) UNIQUE,

password VARCHAR2(10)

);

**Viewing**

Note: Client table is not asked in question but for clientNo in Viewing to make it a foreign key, we’ve to first make a table Client and make clientNo primary key in it.

CREATE TABLE Client(

clientNo VARCHAR2(10) primary key,

fName VARCHAR2(10),

IName VARCHAR2(10),

telNo VARCHAR2(20),

prefType VARCHAR2(10),

maxRent NUMBER(4),

eMail VARCHAR2(20)

);

CREATE TABLE Viewing (

clientNo VARCHAR2(10),

propertyNo VARCHAR2(10),

viewDate DATE,

commment VARCHAR2(30),

PRIMARY KEY(clientNo,propertyNo),

FOREIGN KEY (clientNo) REFERENCES client(clientNo),

FOREIGN KEY (propertyNo) REFERENCES PropertyForRent(propertyNo)

);

**Question #05: Write SQL Queries for each of the following:**

a) Make a list of all property numbers for properties that involve a Female Staff member.

SELECT propertyNo FROM PropertyForRent WHERE staffNo IN

(SELECT staffNo FROM Staff WHERE sex = ‘F’);

b) Retrieve all privateOwners who own a ‘House’

SELECT fName, lName FROM privateOwners WHERE ownerNo IN

(SELECT ownerNo FROM PropertyForRent WHERE type = ‘House’);

c) Show the resulting salaries if every staff member working in ‘London’ is given a 15% raise

SELECT fName, lName , salary AS “Old Salary”, salary+ (salary\*0.15) AS “New Salary” FROM staff WHERE branchNo IN (SELECT branchNo FROM Branch WHERE city = ‘London’);

d) Retrieve all the properties which were viewed in ‘May 2013’

Select \* FROM PropertyForRent WHERE propertyNo IN

(SELECT propertyNo FROM Viewing WHERE to\_char(hire\_date,'Mon-yyyy') = ‘May-2013)’

e) Retrieve all the staff members whose first name starts with ‘D’ or last name starts with ‘W’

SELECT \* FROM Staff where fName LIKE ‘D%’ OR lName LIKE ‘W%’;