Assignment One

Maxwell Reid
s3787033

Date: 08/08/2020

ISYS1057 – Database Concepts

Question 1 – The Beautiful House Case Study

Beautiful House is a real-estate agency that has offices throughout Melbourne. The following are the requirements for managing data about staff, offices and properties for Beautiful House.
Offices

Offices

Beautiful House has branch offices throughout Melbourne. Each branch office is allocated several staff members, including a manager who manages the operations of the office. The data describing a branch office includes a unique branch number, address (street, suburb and postcode), telephone numbers (up to a maximum of three), and the name of the staff member who currently manages the office. Additional data about managers includes the date that the manager started the position at the current branch office and a monthly bonus payment based on performance.

Staff

Staff members with the role of supervisor are responsible for the day-to-day activities of a team of staff called agents (up to a maximum of 10 at any one time). Not all staff members are assigned to a supervisor. The data for each staff member includes a unique staff number, name, address, position, salary, name of supervisor (where applicable), and the details of the branch office where a staff member is working.

Properties

Each branch office offers a range of properties for rent. The data stored for each property includes a unique property number, address (street, suburb, postcode), type, number of rooms and weekly rent. The management of a property is assigned to an agent. An agent manages a maximum of 100 properties at any one time. When a property is available for rent, the property details will be displayed on the Beautiful House website and, when necessary, as advertisement on other websites.
A database system will be built to help manage data for Beautiful House. Answer the following questions. Each answer should be a SHORT passage of at most several lines.

QUESTION 1.1. (2 marks): Discuss two main important advantages for using a DBMS to manage data for Beautiful House and explain the reasons for your choices based on the given scenario.

A database management system is used to efficiently and effectively manage and store data.

Utalising a DBMS allows Beautiful House to share data across their several branches across Melbourne as it can be hosted on a network. This is beneficial as modifications made on one system will replicate across the branches increasing data integrity.

A DBMS will also allow Beautiful House to ensure that their data is secure and private. Through the use of access controls, they are able to limit which employees can access specific sets of data. A good example of this is the details about staff such as address, and salary should not be accessible by employees unless they are authorized.

QUESTION 1.2. (3 marks): Name three classes of objects that need to be represented in the database and details need to be stored in the database. Explain your answer.

Class 1. Offices

Details that need to be stored:
- Unique branch number
- Address (Street, suburb, postcode)
- Telephone numbers (Up to three)
- Managers Name

Class 2. Properties

Details that need to be stored:
- Unique property number
- Address (Street, suburb, postcode)
- Type
- Number of rooms
- Weekly rent
- Agent
- Availability

Class 3. Staff

Details that need to be stored:
- Unique staff number
- Name
- Address
- Position
- Salary
- Supervisor Name (if applicable)
- Branch details
- Role
- Monthly Bonus (if applicable)
- Office Start Date (if applicable)

QUESTION 1.3 (3 marks): Give three relationships you think exist among the classes of objects in Question 1.2. Explain your answer.

Relationship 1. Staff assigned to office
A staff with the role agent is assigned to work at a specific office

Relationship 2. Property assigned to staff
A property is assigned to a staff with the role agent

Relationship 3. Staff assigned to staff
Some staff with the role agent are assigned to a staff with the role manager

QUESTION 1.4. (2 marks): Assume a database system is built and in operation. Give two classes of users of the database system and list their two likely queries.

Agents:

1. Agents may need to query the database system for which of their assigned properties are currently available.
2. Agents may query the database system for properties assigned to them within a given price range.

Managers

1. A manager of a store may complete a query to view the number of properties that are assigned to an agent.
2. A manager may also query for what properties currently have no assigned agent.

Question 2 – Employee Database

This section contains the schema and a database instance for the Employee database that stores employee data for an organisation. The data includes items such as personal info (e.g. name, phone, salary), departments of the organisation (e.g. name and location of each department, who the manager is), jobs (e.g. titles, salary range), and a history for past contracts with each employee. A database instance is shown in Figure 1 followed by the database schema.

Employees

employee_id	**first_name**	**last_name**	**phone_number**	**hire_date**	**job_id**	**salary**
50	Adam	Smith	1234	26/10/09	22	$66,000
66	Tom	Moosa	1235	10/12/16	10	$140,000
10	Jonny	Deans	1236	9/9/15	33	$70,000
12	Adam	Jones	1247	8/8/19	10	$138,000
18	Joseph	Ryan	1277	5/5/20	10	$150,000

Departments

department_id	department_name	manager_id	location_id
1	IT Services	12	10
2	Accounting	66	20
3	Human Resources	18	30

Jobs

job_id	job_title	min_salary	max_salary
10	Dep Manager	$120,000	$150,000
22	Accountant	$60,000	$80,000
33	Programmer	$60,000	$80,000
45	Senior Programmer	$70,000	$120,000

Locations

location_id	street_address	postal_code	city	state_province	country_id
10	123 Collins St	3000	Melbourne	VIC	1
20	222 Bourke St	3000	Melbourne	VIC	1
30	555 Swanston St	3000	Melbourne	VIC	1

Countries

country_id	country_name
1	Australia
2	Vietnam
3	Spain

Job History

employee_id	start_date	end_date	job_id	department_id
10	1/1/01	10/4/02	33	1
10	11/4/02	20/8/02	33	1
12	1/1/98	5/10/03	33	1
12	6/10/03	6/10/04	33	1
12	7/10/04	7/8/09	33	1

The database schema is shown below where attribute meaning is self-explanatory; "Job History" is simplified as "JobHistory". Primary keys are underlined, and foreign keys are annotated with a *.

Employees(employee_id, first_name, last_name, phone_number, hire_date, job_id, salary)
Departments(department_id, department_name, manager_id*, location_id*)
Jobs(job_id, job_title, min_salary, max_salary)
Locations(location_id, street_address, postal_code, city, state_province, country_id*)
Countries(country_id, country_name)
JobHistory(employee_id*, start_date, end_date, job_id*, department_id*)

Foreign Key		Primary Key
Job History.employee_id	-------->	Employees.employee_id
Departments.manager_id	-------->	Employees.employee_id
Job History.department_id	-------->	Departments.department_id
Job History.job_id	-------->	Jobs.job_id
Locations.country_id	-------->	Countries.country_id
Departments.location_id	-------->	Locations.location_id
Employees.department_id	-------->	Departments.department_id

The following questions must be answered based on the given database schema and instance. Where explanation is required, each answer should be a SHORT passage of at most several lines. You are strongly advised to include your SQL scripts for building all tables in the database (schema and instance) as Appendix to show that your SQL statements for Questions 2.6—2.8 can run in Oracle. Note that in Figure 1 some values are represented in currency format for readability; they are indeed of INTEGER data type in Oracle SQL.

QUESTION 2.1 (1 mark): Does the database schema ensure that there is a job associated with each employee? Explain your answer.

No this system does not ensure there is a job associated with each employee as it is possible to set the job_id to null. This is because job_id has not been set as a foreign key that references the job_id in the Jobs table.

QUESTION 2.2 (1 mark): Can an employee take two jobs at the same time? Explain your answer?

With how the DBMS is structured it is not possible for an employee to have multiple jobs. Each employee has only one specified job in the employees table and adding multiple entries of the same employee would not be possible as the employee_id is the primary key.

QUESTION 2.3 (1 mark): The Accounting department has recently changed its structure and now has branches at all locations of the business. The following SQL statements are intended to record this new change in the database instance. Will they work? Explain your answer.

INSERT INTO Departments VALUES(2, 'Accounting', 66, 10);
INSERT INTO Departments VALUES(2, 'Accounting', 66, 30);

These SQL statements will not work with how the current database is configured. This is because department_id is set as the primary key of the table and this means it must be unique within the table. Attempting to add two records with the department_id as 2 violates this constraint as one already exists.

2.4 (1 mark): Consider the request "find all the employment details (e.g. first name, last name, hire date) of department managers". Can this request be completed using the given database schema? Explain your answer.

Yes, it is possible to complete this query by utalising a join on the employee_id of the employee table and the manager_id of the department table.

2.5 (1 mark): Explain what the result of executing the following SQL statement on the database instance will be.

DELETE from JOBS WHERE job_id=33;

It will not be possible to run the SQL statement as there is currently an employee that exists with the job_id of 33. This employee would need to be removed first for the SQL statement to work.

2.6 (2 marks): Write an SQL statement to create the Department table including all the constraints, assuming all the tables that Department depends on already exist in the database. Your SQL statement must execute in Oracle.

2.7 (2 marks): Write an SQL statement to create the JobHistory table including all the constraints, assuming all the tables that JobHistory depends on already exist in the database. Your SQL statement must run in Oracle.

2.8 (1 mark): A new department is recently established with the name of Property Services. You are asked to update the given database instance so that it also includes the new department. The new department is located at 555 Swanston Street Melbourne VIC and its manager is yet to be assigned. Write an SQL statement to complete the request. Your SQL statement must run in Oracle.

INSERT INTO DEPARTMENTS VALUES (4,'Property Services',null,30);

Appendix A

```
CREATE TABLE Employees(
employee_id int,
first_name varchar(30),
last_name varchar(30),
phone_number char(4),
hire_date date,
job_id int,
salary int,
CONSTRAINT pk_employee_id PRIMARY KEY (employee_id)
);

INSERT INTO Employees VALUES (50,'Adam','Smith','1234','26/OCT/2009',22,66000);
INSERT INTO Employees VALUES (66,'Tom','Moosa','1235','10/DEC/2016',10,140000);
INSERT INTO Employees VALUES (10,'Jonny','Deans','1236','09/SEP/2015',33,70000);
INSERT INTO Employees VALUES (12,'Adam','Jones','1247','08/AUG/2019',10,138000);
INSERT INTO Employees VALUES (18,'Joseph','Ryan','1277','05/MAY/2020',10,150000);

CREATE TABLE Jobs(
job_id int,
job_title varchar(30),
min_salary int,
max_salary int,
CONSTRAINT pk_job_id PRIMARY KEY (job_id)
);

INSERT INTO Jobs VALUES (10,'Deb Manager',120000,150000);
INSERT INTO Jobs VALUES (22,'Accountant',60000,80000);
INSERT INTO Jobs VALUES (33,'Programmer',60000,80000);
INSERT INTO Jobs VALUES (45,'Senior Programmer',70000,120000);

CREATE TABLE Countries(
country_id int,
country_name varchar(30),
CONSTRAINT pk_country_id PRIMARY KEY (country_id)
);

INSERT INTO Countries VALUES (1, 'Australia');
INSERT INTO Countries VALUES (2, 'Vietnam');
INSERT INTO Countries VALUES (3, 'Spain');

CREATE TABLE Locations(
location_id int,
street_address varchar(30),
postal_code varchar(4),
```

```
city varchar(30),
state_province varchar(4),
country_id int,
CONSTRAINT pk_location_id PRIMARY KEY (location_id),
CONSTRAINT fk_country_id FOREIGN KEY (country_id) REFERENCES Countries (country_id)
);

INSERT INTO Locations VALUES (10,'123 Collins St','3000','Melbourne','VIC',1);
INSERT INTO Locations VALUES (20,'222 Bourke St','3000','Melbourne','VIC',1);
INSERT INTO Locations VALUES (30,'555 Swanston St','3000','Melbourne','VIC',1);

CREATE TABLE Departments(
department_id int,
department_name varchar(30),
manager_id int,
location_id int,
CONSTRAINT pk_department_id PRIMARY KEY (department_id),
CONSTRAINT fk_manager_id FOREIGN KEY (manager_id) REFERENCES Employees (employee_id),
CONSTRAINT fk_location_id FOREIGN KEY (location_id) REFERENCES Locations (location_id)
);

INSERT INTO Departments VALUES (1,'IT Services',12,10);
INSERT INTO Departments VALUES (2,'Accounting',66,20);
INSERT INTO Departments VALUES (3,'Human Resources',18,30);

CREATE TABLE JobHistory(
employee_id int,
start_date date,
end_date date,
job_id int,
department_id int,
CONSTRAINT pk_composite PRIMARY KEY (employee_id, start_date, end_date),
CONSTRAINT fk_employee_id FOREIGN KEY (employee_ID) REFERENCES Employees (employee_id),
CONSTRAINT fk_job_id FOREIGN KEY (job_id) REFERENCES Jobs (job_id),
CONSTRAINT fk_department_id FOREIGN KEY (department_id) REFERENCES Departments (department_id)
);

INSERT INTO JobHistory VALUES (10,'1/JAN/2001','10/APR/2002',33,1);
INSERT INTO JobHistory VALUES (10,'11/APR/2002','20/AUG/2002',33,1);
INSERT INTO JobHistory VALUES (12,'1/JAN/1998','5/OCT/2003',33,1);
INSERT INTO JobHistory VALUES (12,'6/OCT/2003','6/OCT/2004',33,1);
INSERT INTO JobHistory VALUES (12,'7/OCT/2004','7/AUG/2009',33,1);
```