Table of Content

|  |  |
| --- | --- |
| A picture of a winding road and trees  database coursework 5ci022  DATABASE FOR RESIDENTIAL CARE HOME | Zahran Mahmood 2100516  Makwembere Allan F. 2108418  DATABASE PROJECT |

ANALYSIS………………………………………………………………………………………………………………………………….2

ERD DIAGRAM………………………………………………………………………………………………………………………….3

TABLE HEADINGS………………………………………………………………………………………………………………………4

TABLE CREATION……………………………………………………………………………………………………………………….5

TABLE DATA……………………………………………………………………………………………………………………………….8

SQL & QUERRIES…………………………………………………………………………………………………………………………13

DML COMMANDS……………………………………………………………………………………………………………………….16

# Analysis

The subject business is a mental-health care home that has been recently set up and it requires a database system to assist its employees in helping the service users/Residents. The database will therefore store personal data about the employees assisting the residents and likewise store personal data about the residents receiving the assistance.

This data is useful for equipping the employees with the relevant information to allow for organised appointment setting for each resident who requires the assistance of an employee. Through the usage of the database the employee can plan for their workday, monitor their job role and responsibility, monitor any appointments they have, acquire all the information they need for clients and manage their allocated Resident’s wellbeing and finances easily.

To meet this requirement the database must store data about:

* The employees and details.
* The residents and their conditions.
* The interaction times and schedules of the Residents and Employees.
* The emergency contacts of the Residents in case of emergency
* The finances of the Residents under the management of the Employees
* The job roles and managers in charge of their implementation.

For the above information to be stored there needs to be a strong layer of security to protect the privacy and confidentiality of the parties involved. A lot of the queries used would be accessing private personal information. Since this data is classified as special category data, there needs to be mutual explicit consent that would permit Database administrators to access this data along with confidential handling of this data. All according to the GENERAL DATA PROTECTION REGULATION (2018).

The data must be lawful, fair and transparent. It must be for the specific purpose it is needed. It must be adequate and must not store unnecessary data. It must be accurate according to the factual details and up to date. It must be stored safely and for the period it is in use and discarded safely once it’s no longer needed. It must also be erased easily in case it’s no longer needed.

In keeping with this several business rules were agreed upon to ensure that the database works efficiently. These are:

* Each employee must belong to a department and must be identified uniquely.
* Every employee must have a valid email address registered to gmail.
* All records must be stored in uppercase.
* Since there only 9 flats, Resident identifiable flat numbers must be less than 10.
* Residents must be above 18 and below 60 years of age.
* The private finances of the employees should only be handled by the 2 allocated care support staff.
* The amount spent from the Resident’s accounts must not exceed £100.
* Appointments for the Residents can only be given to, Support Workers, Social Workers, Psychologists and Therapists.

# ERD DIAGRAM

Diagram

Description automatically generated

# ASSUMPTIONS

* We will allow every employee to have an appointment with a Resident by booking it through the appointment table, here the RESNO AND EMPNO will be used to identify an appointment.
* The finance can only be handled by the allocated employees in the Resident Finance table.
* Every Resident can have as many Contacts/Next of Kins as they wish.

# TABLE HEADINGS

**DEPARTMENT**(DEPTNO, DEPTNAME )

**EMPLOYEE**(EMPNO,  EMPNAME,   \*DEPTNO,   JOB ,  SAL ,  EMAIL)

**RESIDENT**(RESNO,  RESNAME,  DOB ,  age,  CONDITION,  FLATNO)

**RESIDENTFINANCE**(\*RESNO, \*EMPNO,  BALANCE,  REFERENCE\_NUM,  TRANSDATE, AMOUNT\_OUT,  AMOUNT\_IN)

**CONTACT**(\*RESNO, NEXTOFKIN, PHONENUMBER,  ADDRESS)

**APPOINTMENT**(\*EMPNO, \*RESNO, AP\_DATE, ACTIVITY)

# Table Creation

In order to create the table, we used Notepad+ to write the SQL DDL commands and then linked the code to oracle. The constraints have been defined to be table level and not just for individual columns.

Our Code

Department Table

Text

Description automatically generated

Employee Table

A screenshot of a computer

Description automatically generated

Resident Table

Text

Description automatically generated

ResidentFinance Table

Graphical user interface

Description automatically generated

Contact Table

Text

Description automatically generated

Appointment Table

Text

Description automatically generated

# Table Data

Again, to populate the tables we made use of the Notepad+ in order to fill in the DML SQL data. This was also linked to Oracle.

EMPLOYEE TABLE

A screenshot of a computer screen

Description automatically generated

RESIDENT TABLE

Graphical user interface, text

Description automatically generated

APPOINTMENT TABLE

Graphical user interface

Description automatically generated

RESIDENTFINANCE TABLE

Graphical user interface, text

Description automatically generated

CONTACT TABLE

A screenshot of a computer

Description automatically generated with medium confidence

DEPARTMENT TABLE

Text

Description automatically generated with medium confidence

# SQL AND QUERRIES

F) Here we will test the use of the BETWEEN logical operator.

Graphical user interface

Description automatically generated

The above SQL looks through the JOINED EMPLOYEE and APPOINTMENT tables for an employee that has a scheduled appointment with a resident who has a RESNO(resident number) BETWEEN 601 and 603 whilst not ignoring any rows that do not satisfy the join condition by use of the OUTER JOIN(+).

G) Here we test the use of the DISTINCT function.

Graphical user interface

Description automatically generated

The above SQL selects the distinct (or different) RESNAME (resident names) from the RESIDENT table without any duplicate values.

H) Here we test the TO\_CHAR function.

Graphical user interface, application

Description automatically generated

Here we select the AP\_DATE (appointment date) and use the TO\_CHAR function to change the format of the date to be shown in full including the day of the week and for it all to be shown in French.

1. Here we show the aggregate function.

Graphical user interface

Description automatically generated

We use the MIN aggregate function to show the minimum FLATNO (flatno) from the resident table and name the result of the query as THE\_FIRST\_ROOM\_VACANCY.

J) The following shows the GROUP BY command:

Th following statement uses the aggregate function of MIN() to show the minimum balance and then groups all the rows that have the same minimum amount returned into the same rows grouped according to their EMPNO.

A screenshot of a computer

Description automatically generated

K) The following shows the use of a SUBQUERY.

The query selects Resident details from two different tables, CONTACT and RESIDENT and the proceeds to match it up according to their respective NEXTOFKIN’s on the condition that they have more than one contact or NEXTOFKIN shown by a repeating RESNO which is selected by the subquery.

A screenshot of a computer

Description automatically generated

# DML Commands

L) DELETE command testing the FOREIGN KEYS.

Graphical user interface, application

Description automatically generated

The above query does not work because it is trying to delete a parent record that has a child record or foreign key in another table, namely the CONTACT, RESIDENTFINANCE and APPOINTMENT tables.

Graphical user interface, application

Description automatically generated

The above query works because it is deleting a child record that has little referential integrity except to its parent.

M) UPDATE command testing the FOREIGN KEYS.

The query below generates an error because there is no relationship to any record or parent keys that correspond to the newly set EMPNO and RESNO which were declared as foreign keys in the constraints.

Graphical user interface, application

Description automatically generated

The one below also fails to execute due to it having child records in other tables.

Graphical user interface, application

Description automatically generated

The query below works because it updates the ACTIVITY and AP\_DATE columns which have no parent or child records as they are not keys. So it is impossible to update keys that have foreign keys in other tables and one must first delete the child records or tables in order to update in the parent tables or record.

Graphical user interface, application

Description automatically generated

N) INSERT command testing the PRIMARY KEYS.

The query below fails to execute because it contains a familiar EMPNO as a primary key that is defined for another existing record thereby violating the PRIMARY KEY constraint that only uses unique identifiers for each record.

A screenshot of a computer

Description automatically generated`

However, the query below works because it’s Primary Key or EMPNO is unique and does not identify another record showing the importance of uniqueness in the creation of primary keys to avoid duplication and allow for easier record identification.

Graphical user interface, application

Description automatically generated

# PEER REVIEW

The database sections were done by both of us.

1. Analysis- Done by Allan F Makwembere
2. ERD DIAGRAM- Done by Zahran Mahmood
3. Assumptions- Done by Allan F Makwembere
4. Table Headings- Done by Zahran Mahmood
5. Table Creation- Done by Allan F Makwembere
6. Table Data- Done by both of us.
7. SQL AND QUERIES- Done by Zahran Mahmood and some by Allan F Makwembere
8. DML Commands- Done by Allan Makwembere.

Zahran signature:

Graphical user interface, application, Word

Description automatically generated

Allan signature: Graphical user interface, application, Word

Description automatically generated