Database Systems & Security

Patient Appointment Management System

1.0 **Introduction**

A Patient Appointment Management System is a system with an application interface that helps the Surgery uniquely manage their patients' information. The need for patient information management arises from the requirement to access patient data across various levels, including diagnosis and service payment within the healthcare facility.

A patient information system plays a vital part in improving the patient experience, and the surgery’s workflow, and optimizing resource allocations.

The advantages include:

- Enhanced communication between patients and staff
- Decreased wait time
- Better record-keeping
- Expedited administrative task
- Increased protection of patient data by limiting unauthorized access

1.1 **Information Needs and Data Involved**

In this database design, the information needs revolve around efficiently scheduling and managing appointments, ensuring effective communication between patients and healthcare staff, and maintaining accurate records. The data involved encompass various aspects of patient appointments, including personal details, appointment dates and times, medical history, treatment plans, and follow-up schedules.

In the patient appointment management system, appointments must be scheduled based on both patient availability and healthcare provider availability, ensuring optimal resource allocation. Additionally, appointment cancellations and rescheduling requests should be accommodated within a reasonable timeframe to minimize disruptions and optimize appointment slots.

1.2 **Data Requirement**

The data required for the design of this system includes:

- Patient data: This is important for basic personal details, like the name, date of birth, and address.
- Appointment records: Past, present, and future appointment dates and times are necessary to assist both patients and staff in monitoring and tracking schedules accurately.
- Medical records: The patient's medical information from every appointment held is stored for future diagnosis and planning improvements in treatment.
- Staff: records of staff are maintained who attend to patients on appointment days are stored. Data needs include staff name, contact email, specialty, and department.

2.0 **Data Modelling**

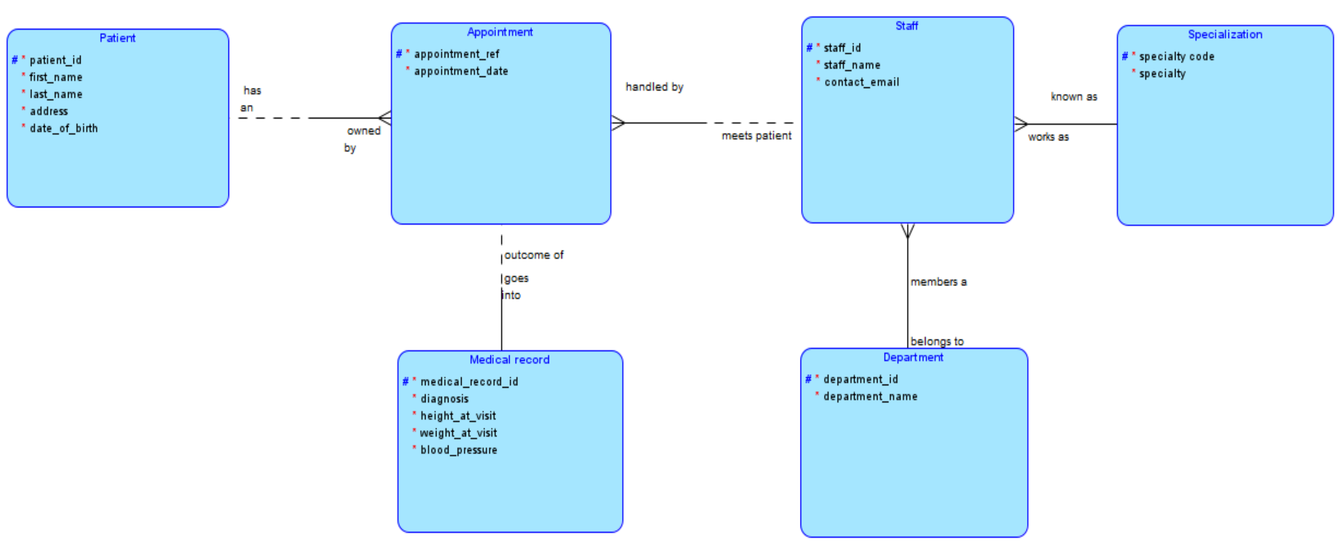
Data Modelling is the processing of defining the structure of your database or information system. This provides a broad understanding of how required data would be structured, including potential entities, attributes, and relationships between these entities.

Data modelling is an important step in building this information management system for the following reasons;

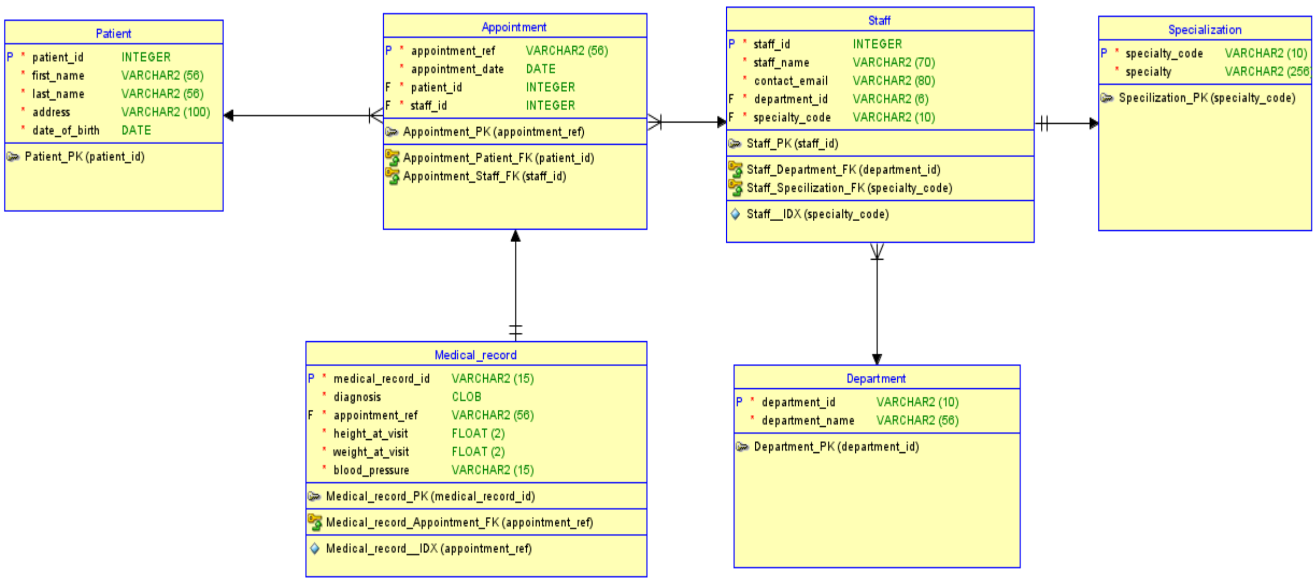
1. Data modelling provides a clear understanding of how the required data will be organized within the system. Progressing through all stages of data modelling, from conceptual to logical to physical models, offers clarity to both administrators and clients and improves data quality by identifying gaps in the system, entities, relationships, and information needs that should not be missed
2. The ERD model is the blueprint of the schema; providing guidance for data retrieval from any entity for analytical purposes and offering clarity on entity relationships.
3. It reduces redundancy by preventing the replication or repetition of the same information in multiple locations, thereby preserving the integrity of the data.
4. It enhances data quality by pinpointing gaps in the system, entities, relationships, and information requirements that must not be overlooked.

2.1 **Entity Relationship Diagram (ERD) Model**

The logical model for the patient information management system was drawn using the Oracle Data Modeller. It shows the entities, the relationship between entities, their attributes, datatype, and cardinalities. The physical model expresses the logical model in the way the data would be in the database. It shows not just the diagram relationship between entities but also the actual data attribute relationship expressed as foreign keys. It indicates the primary keys with the denotation P and foreign keys by F.



Engineered to



2.2 **Entities and Attributes**

The entities and attributes drawn up in the models are;

1. Patient: This entity contains information on any patient registered with the surgery. This information includes the patient's name, gender, address, and date of birth.
2. Appointment: This entity includes records of scheduled appointments within the system, storing relevant details including appointment date, time, associated patient, and staff member assigned to the appointment.
3. Medical record: stores all past diagnoses made during an appointment, treatment plan and medications. This means that for any appointment held, a corresponding medical record is created to store the medical information and diagnosis of the patient.
4. Staff: The staff entity uniquely identifies all staff that work with the surgery, it contains data on staff such as staff name, and contact email.
5. Department: Every staff in the surgery is employed under a department. The department entity stores data on the staff and the department they belong.
6. Specialization: This entity provides information about the specific expertise or area of focus of the staff working within the surgery. It assists in identifying the particular role or medical discipline of each staff member within the surgery.

2.3 **Cardinalities and Relationships**

In database management, a relationship refers to how entities are related. It tells the connection between entities. Cardinality, on the other hand, describes the number of members of an entity that can be linked with another member of an entity. The diagram illustrates how many members from each group are linked to a member in another group. There are various types of cardinalities, including one-to-one, one-to-many, and many-to-many.

Below are the relationships and cardinalities of this database:

- Patient to Appointment: The "Patient" and "Appointment" entities exhibit a one-to-many cardinality. It is optional for a patient to have an appointment, a patient can have multiple appointments, and each appointment is uniquely owned by the patient.
- Appointment to Medical Record: The "Appointment" and "Medical Record" entities have a one-to-one cardinality. Each medical record corresponds to an appointment held. This relationship is optional for appointments, as only held appointments have a corresponding medical record stored.
- Appointment to staff: The "Appointment" and "Staff" entities have a many-to-one cardinality. This indicates that each appointment is associated with one staff member, while a staff member can be linked to multiple appointments, either held or scheduled. It is also possible that a staff does not have an associated appointment.
- Staff to Department: Every staff belongs to a department and a department can have one or more members of staff. The cardinality between the two entities is a many-to-one where a staff belongs to only one department and a department can have multiple staff members.
- Staff to Specialization: Each staff member operates within a specific specialization within the surgery and is identified by his or her specialization. The "Staff" and "Specialization" entities have a many-to-one cardinality. One or more staff can be a doctor or a nurse.

3.0 **Normalization**

Normalization in database management focuses on organizing data in relational databases to reduce redundancy and reliance, ensuring data integrity and consistency. It involves structuring tables and attributes to minimize data duplication and anomalies, thereby optimizing storage efficiency and enhancing database performance (Elmasri & Navathe, 2015).

Normalization upholds data integrity and consistency by eliminating or minimizing data redundancy, which helps to prevent anomalies and ensures that each data element is stored only once. It also simplifies database design and management, making it easier to understand and maintain the database schema over time (Connolly & Begg, 2014).

To begin the process of normalizing the data for our information system design, we commence with the raw data required for the surgery patient information management system as shown below:

patient_id	first_name	last_name	date_of_birth	address	appointmen	appointment_date	medical_rec	diagnosis	height_at_v	weight_at_v	blood_press	staff_i	staff_name	contact_em	department_i	department	speciality	speciality_co
34	Loveline	Johnson	24/02/1986	30 Sagba Road	APT-1460	19/05/2024 12:30						347	Jack Jones	jjones.992@g	Obstetrics & G	Obste-H	Doctor	DD-1
27	Kim	Helen	22/02/1996	52 Park Lane	APT-3086	17/05/2024 10:45						366	Fiona Naey	fnaey.592@g	Pediatrics	Pedia-H	Doctor	DD-1
61	Due	Miller	25/02/1983	63 Avenue street	APT-2249	15/04/2024 10:00						366	Fiona Naey	fnaey.592@g	Pediatrics	Pedia-H	Doctor	DD-1
24	Niely	Naey	05/03/1948	40 St Peter Road	APT-6086	10/01/2024 14:00	MR-98770	Anaemia	1.7700516	63.918934	98/82 mmHg	394	Henry hdeen	hhdeen.317@	Internal Medici	Inter-H	Gynaecolog	GG-1
60	Praise	Jones	22/02/1994	46 Pepple Creek	APT-8008	09/01/2024 11:45	MR-10145	Acute chest	1.7655683	97.90731	111/71 mmH	394	Henry hdeen	hhdeen.317@	Internal Medici	Inter-H	Gynaecolog	GG-1
38	ThankGod	Naey	06/03/1944	1 Forth and Back	APT-3241	02/01/2024 11:15	MR-51164	Internal ble	1.7663997	75.369346	130/66 mmH	176	Kirch Johnson	kjohnson.908	Pediatrics	Pedia-H	Nurse	NN-1
60	Praise	Jones	22/02/1994	46 Pepple Creek	APT-3344	28/01/2023 12:15	MR-79038	Routine che	1.7691068	87.395381	108/86 mmH	82	George Jones	gjones.570@	Obstetrics & G	Obste-H	Gynaecolog	GG-1
38	ThankGod	Naey	06/03/1944	1 Forth and Back	APT-4077	24/01/2023 09:15	MR-35101	High sugar	1.9827854	79.289156	131/71 mmH	547	Bob Lime	blime.329@g	Internal Medici	Inter-H	Nurse	NN-1

3.1 **First Normal Form (1NF)**

The first normalization rule states that

1. Each record must be unique with no duplicate row.
2. A primary key must be identified

To satisfy this rule,

- Repeating rows refer to records that could potentially be stored in multiple columns but are instead contained within a single column. Upon scanning the table, no repeating rows were present.
- Possible primary keys have been identified as patient_id and appointment_ref. These keys, uniquely determine other data points in the table establishing them as functional dependencies. For instance, patient_id uniquely identifies the patient's name, and address. Similarly, appointment_id identifies appointment_date, medical records, the staff name, department, and specialization.
  The appointment_ref and patient_id are prime attributes that form the primary key.

3.2 **Second Normal Form (2NF)**

The second normalization rule states that

1. The table must satisfy 1NF
2. No partial dependencies - partial dependencies are attributes that depend only on a part of the primary key. In this case, the primary keys (prime attributes) are patient_id and appointment_ref.

To begin, the functional dependencies and partial dependencies are listed. Ensure the resulting table has no partial dependencies. This is done as follows;

3.2.1 **Functional dependencies:** The functional dependencies are

- appointment_ref, patient_id -> first_name, last_name, date_of_birth, address, appointment_date, medical record id, diagnosis, height_at_visit, weight_at_visit, blood_pressure, staff_id, staff_name, email, department_id, department_name, speciality_code, and speciality

3.2.2 **Partial dependencies:**

- **Appointment_ref** -> appointment_date, medical_record_id, diagnosis, height_at_visit, weight_at_visit, blood_pressure, staff_id, staff_name, email, department_id, department_name, speciality_code, and speciality
- **Patient_id** -> first_name, last_name, date_of_birth, and address

Convert from 1NF -> 2NF; remove all partial dependencies

1. Patient table

patient_id	first_name	last_name	date_of_birth	address
34	Loveline	Johnson	24/02/1986	30 Sagba Road
27	Kim	Helen	22/02/1996	52 Park Lane
61	Due	Miller	25/02/1983	63 Avenue street
24	Niely	Naey	05/03/1948	40 St Peter Road
60	Praise	Jones	22/02/1994	46 Pepple Creek
38	ThankGod	Naey	06/03/1944	1 Forth and Back Street
33	George	Wilson	01/03/1967	7 Oakland Off Ladg

2. Appointment table

appointment_ref	appointment_date	medical_record_id	diagnosis	height_at_visit	weight_at_visit	blood_pressure	staff_id	staff_name	contact_email	department_name	department_id	speciality	speciality_code
APT-6086	10/01/2024 14:00	MR-98770	Anaemia	1.770051612	63.91893423	98/82 mmHg	394	Henry hdeen	hhdeen.317@goodhealth.com	Internal Medicine	Inter-H	Gynaecologist	GG-1
APT-8008	09/01/2024 11:45	MR-10145	Acute chest pain	1.765568273	97.90731017	111/71 mmHg	394	Henry hdeen	hhdeen.317@goodhealth.com	Internal Medicine	Inter-H	Gynaecologist	GG-1
APT-3241	02/01/2024 11:15	MR-51164	Internal bleeding fron	1.766399663	75.36934625	130/66 mmHg	176	Kirch Johnson	kjohnson.908@goodhealth.com	Pediatrics	Pedia-H	Nurse	NN-1
APT-3344	28/01/2023 12:15	MR-79038	Routine checkup, all v	1.769106761	87.39538116	108/86 mmHg	82	George Jones	gjones.570@goodhealth.com	Obstetrics & Gynecology	Obste-H	Gynaecologist	GG-1
APT-4077	24/01/2023 09:15	MR-35101	High sugar level	1.982785409	79.28915571	131/71 mmHg	547	Bob Lime	blime.329@goodhealth.com	Internal Medicine	Inter-H	Nurse	NN-1
APT-5226	22/01/2023 09:30	MR-33268	Lower leg swelling, pc	1.592154391	52.07796622	112/72 mmHg	524	George Miller	georgemiller.139@goodhealth.com	Cardiology	Cardi-H	Surgeon	SS-1
APT-2395	19/01/2023 09:00	MR-10075	Minor skin irritation,	1.749234369	55.10309036	91/83 mmHg	524	George Miller	georgemiller.139@goodhealth.com	Cardiology	Cardi-H	Surgeon	SS-1
APT-8967	13/01/2023 15:00	MR-17004	indisgestion	1.672985424	81.02997676	92/89 mmHg	176	Kirch Johnson	kjohnson.908@goodhealth.com	Pediatrics	Pedia-H	Nurse	NN-1

3.3 **Third Normal Form (3NF)**

The third normal form rule states that:

1. The table must satisfy the 2NF rule
2. No transitive dependencies - transitive dependencies are non-key columns that depend on other non-key columns. To satisfy this rule, all transitive dependencies must be removed.

3.3.1 **Transitive dependency**: Identified transactive dependencies are;

Department_id -> department_name
speciality _code -> speciality
staff_id -> staff_name, contact_email
medical record_id -> diagnosis, height_at_visit, weight_at_visit, and blood_pressure

To satisfy the 3NF rule, all transitive dependencies are completely removed and turned into tables.

Staff table

staff_id (pk) -> staff_name, contact_email, department_id (fk), speciality_code (fk)

PK			FK	FK
staff_id	staff_name	contact_email	department_id	speciality_code
347	Jack Jones	jjones.992@goodhealth.com	Obste-H	DD-1
366	Fiona Naey	fnaey.592@goodhealth.com	Pedia-H	DD-1
394	Henry hdeen	hhdeen.317@goodhealth.com	Inter-H	GG-1
176	Kirch Johnson	kjohnson.908@goodhealth.com	Pedia-H	NN-1
82	George Jones	gjones.570@goodhealth.com	Obste-H	GG-1
547	Bob Lime	blime.329@goodhealth.com	Inter-H	NN-1

Medical record table

medical_record_id (pk) -> diagnosis, height_at_visit, weight_at_visit, blood_pressure, appointment_ref (fk)

PK					FK
medical_record_id	diagnosis	height_at_visit	weight_at_visit	blood_pressure	appointment_ref
MR-98770	Anaemia	1.770051612	63.91893423	98/82 mmHg	APT-6086
MR-10145	Acute chest pain	1.765568273	97.90731017	111/71 mmHg	APT-8008
MR-51164	Internal bleeding from accident	1.766399663	75.36934625	130/66 mmHg	APT-3241
MR-79038	Routine checkup, all vital signs normal	1.769106761	87.39538116	108/86 mmHg	APT-3344
MR-35101	High sugar level	1.982785409	79.28915571	131/71 mmHg	APT-4077
MR-33268	Lower leg swelling, possible infection	1.592154391	52.07796622	112/72 mmHg	APT-5226

Appointment_table

Appointment_ref (pk) -> appointment_date, medical_record_id (fk), patient_id (fk), staff_id (fk)

PK		FK	FK
appointment_ref	appointment_date	patient_id	staff_id
APT-1460	19/05/2024 12:30	34	347
APT-3086	17/05/2024 10:45	27	366
APT-2249	15/04/2024 10:00	61	366
APT-6086	10/01/2024 14:00	24	394
APT-8008	09/01/2024 11:45	60	394
APT-3241	02/01/2024 11:15	38	176

Specialization table

speciality_code (pk) -> speciality

PK	
speciality_code	speciality
DD-1	Doctor
GG-1	Gynaecologist
NN-1	Nurse
SS-1	Surgeon
MM-1	Midwife

Department table

Department_id (pk) -> department_name

PK	
department_id	department_name
Obste-H	Obstetrics & Gynecology
Pedia-H	Pediatrics
Inter-H	Internal Medicine
Cardi-H	Cardiology

Patient table

patient_id (pk) -> first_name, last_name, address, date_of_birth

PK				
patient_id	first_name	last_name	date_of_birth	address
34	Loveline	Johnson	24/02/1986	30 Sagba Road
27	Kim	Helen	22/02/1996	52 Park Lane
61	Due	Miller	25/02/1983	63 Avenue street
24	Niely	Naey	05/03/1948	40 St Peter Road
60	Praise	Jones	22/02/1994	46 Pepple Creek
38	ThankGod	Naey	06/03/1944	1 Forth and Back Street

3.4 Importance of Normalization and Criteria for Assessing Table Normalization

1. **Redundancy**: Before normalization, there was a significant amount of repetition in the table, with department names, patient names, and staff specializations appearing multiple times. However, after normalization, each table contains unique records of data. For instance, the specialization table now contains unique records for each staff specialization, and the department table contains unique department names.
2. **Anomalies:** Due to the normalization, these anomalies have been avoided. The following explains these anomalies;
   - Update anomalies: Before normalization, if a staff member was moved to another department, updating this information would require modifying multiple records, leading to potential inconsistencies. However, normalization ensures that by updating the staff's department_id in the staff table, the staff is automatically linked to the corresponding new department in the department table by the department_id (foreign key).
   - Delete Anomalies: Without normalization, deleting a unique staff record, such as the only cardiologist, would result in the deletion of all attributes associated with that staff member, potentially reducing the number of specializations. However, normalization prevents this by ensuring that deleting a staff record from the staff table does not affect other related tables, such as the department table.

- Insert Anomalies: Before normalization, adding a new patient record might leave other columns null, as not every patient necessarily has an appointment. With a normalized table, such as in the patient table, new patient data can be safely added without affecting other entities.

3. **Integrity Constraints:** During normalization, foreign keys were introduced to enforce referential integrity and primary keys to maintain entity constraints. As seen in the staff table, department_id and speciality_code are foreign keys referencing the department and specialization table respectively.

4.0 **Database Implementation**

The model created is implemented by creating the tables, inserting data and answering some business questions using SQL.

4.1 **Table Creation:** To create tables for the information management system, Oracle SQL format was utilized. The scripts can be found below;

```
CREATE TABLE patient (
    patient_id    INTEGER NOT NULL PRIMARY KEY,
    first_name    VARCHAR2(56) NOT NULL,
    last_name     VARCHAR2(56) NOT NULL,
    address       VARCHAR2(100) NOT NULL,
    date_of_birth DATE NOT NULL
)
```

Table created.

```
CREATE TABLE department (
    department_id   VARCHAR2(10) NOT NULL PRIMARY KEY,
    department_name VARCHAR2(56) NOT NULL
)
```

Table created.

```
CREATE TABLE specialization (
    specialty_code  VARCHAR2(10) NOT NULL PRIMARY KEY,
    specialty       VARCHAR2(256) NOT NULL
)
```

Table created.

```
CREATE TABLE staff (
    staff_id       INTEGER NOT NULL PRIMARY KEY,
    staff_name     VARCHAR2(70) NOT NULL,
    contact_email  VARCHAR2(80) NOT NULL,
    department_id  VARCHAR2(6) NOT NULL,
    specialty_code VARCHAR2(10) NOT NULL,
    FOREIGN KEY (department_id) REFERENCES department(department_id) ON DELETE CASCADE,
    FOREIGN KEY (specialty_code) REFERENCES specialization(specialty_code) ON DELETE CASCADE
)
```

Table created.

```sql
CREATE TABLE appointment (
    appointment_ref  VARCHAR2(56) NOT NULL PRIMARY KEY,
    appointment_date TIMESTAMP(0) NOT NULL,
    patient_id       INTEGER NOT NULL,
    staff_id         INTEGER NOT NULL,
    FOREIGN KEY (patient_id) REFERENCES patient(patient_id) ON DELETE CASCADE,
    FOREIGN KEY (staff_id) REFERENCES staff(staff_id) ON DELETE CASCADE
)
```

Table created.

```sql
CREATE TABLE medical_record (
    medical_record_id VARCHAR2(15) NOT NULL PRIMARY KEY,
    diagnosis         CLOB NOT NULL,
    appointment_ref   VARCHAR2(56) NOT NULL,
    height_at_visit   FLOAT(2) NOT NULL,
    weight_at_visit   FLOAT(2) NOT NULL,
    blood_pressure    VARCHAR2(15) NOT NULL,
    FOREIGN KEY (appointment_ref) REFERENCES appointment(appointment_ref) ON DELETE CASCADE
)
```

Table created.

4.2 **Data Insertion**

To insert data into the created tables, SQL **INSERT ALL** format was used to insert multiple data into the tables. The scripts are as shown below;

```sql
INSERT ALL
INTO specialization (specialty_code,specialty) VALUES ('DD-1','Doctor')
INTO specialization (specialty_code,specialty) VALUES ('GG-1','Gynaecologist')
INTO specialization (specialty_code,specialty) VALUES ('NN-1','Nurse')
INTO specialization (specialty_code,specialty) VALUES ('SS-1','Surgeon')
SELECT 1 FROM DUAL
```

4 row(s) inserted.

```sql
INSERT ALL
INTO department (department_id,department_name) VALUES ('Obste-H','Obstetrics & Gynecology')
INTO department (department_id,department_name) VALUES ('Pedia-H','Pediatrics')
INTO department (department_id,department_name) VALUES ('Inter-H','Internal Medicine')
INTO department (department_id,department_name) VALUES ('Cardi-H','Cardiology')
SELECT 1 FROM DUAL
```

4 row(s) inserted.

```sql
INSERT ALL
INTO staff (staff_id, staff_name, contact_email, department_id, specialty_code) VALUES (347,'Jack Jones','jjones.992@goodhealth.com','Obste-H','DD-1')
INTO staff (staff_id, staff_name, contact_email, department_id, specialty_code) VALUES (366,'Fiona Naey','fnaey.592@goodhealth.com','Pedia-H','DD-1')
INTO staff (staff_id, staff_name, contact_email, department_id, specialty_code) VALUES (394,'Henry hdeen','hhdeen.317@goodhealth.com','Inter-H','GG-1')
INTO staff (staff_id, staff_name, contact_email, department_id, specialty_code) VALUES (176,'Kirch Johnson','kjohnson.908@goodhealth.com','Pedia-H','NN-1')
INTO staff (staff_id, staff_name, contact_email, department_id, specialty_code) VALUES (82,'George Jones','gjones.570@goodhealth.com','Obste-H','GG-1')
INTO staff (staff_id, staff_name, contact_email, department_id, specialty_code) VALUES (547,'Bob Lime','blime.329@goodhealth.com','Inter-H','NN-1')
INTO staff (staff_id, staff_name, contact_email, department_id, specialty_code) VALUES (524,'George Miller','georgemiller.139@goodhealth.com','Cardi-H','SS-1')
INTO staff (staff_id, staff_name, contact_email, department_id, specialty_code) VALUES (42,'Loveline hdeen','lhdeen.749@goodhealth.com','Inter-H','DD-1')
INTO staff (staff_id, staff_name, contact_email, department_id, specialty_code) VALUES (673,'Charlie Smith','csmith.985@goodhealth.com','Obste-H','NN-1')
SELECT 1 FROM DUAL
```

9 row(s) inserted.

```sql
INSERT ALL
INTO patient (patient_id,first_name,last_name,date_of_birth,address) VALUES (34,'Loveline','Johnson',TO_DATE('24-02-1986', 'dd-MM-yyyy'),'30 Sagba Road')
INTO patient (patient_id,first_name,last_name,date_of_birth,address) VALUES (27,'Kim','Helen',TO_DATE('22-02-1996', 'dd-MM-yyyy'),'52 Park Lane')
INTO patient (patient_id,first_name,last_name,date_of_birth,address) VALUES (61,'Due','Miller',TO_DATE('25-02-1983', 'dd-MM-yyyy'),'63 Avenue street')
INTO patient (patient_id,first_name,last_name,date_of_birth,address) VALUES (24,'Niely','Naey',TO_DATE('05-03-1948', 'dd-MM-yyyy'),'40 St Peter Road')
INTO patient (patient_id,first_name,last_name,date_of_birth,address) VALUES (60,'Praise','Jones',TO_DATE('22-02-1994', 'dd-MM-yyyy'),'46 Pepple Creek')
INTO patient (patient_id,first_name,last_name,date_of_birth,address) VALUES (38,'ThankGod','Naey',TO_DATE('06-03-1944', 'dd-MM-yyyy'),'1 Forth and Back  Street')
INTO patient (patient_id,first_name,last_name,date_of_birth,address) VALUES (33,'George','Wilson',TO_DATE('01-03-1967', 'dd-MM-yyyy'),'7 Oakland Off Ladg')
INTO patient (patient_id,first_name,last_name,date_of_birth,address) VALUES (72,'Isabella','Wisdom',TO_DATE('03-03-1956', 'dd-MM-yyyy'),'4 Harry Smiles Road')
SELECT 1 FROM DUAL
```

8 row(s) inserted.

```sql
INSERT ALL
INTO appointment (appointment_ref,appointment_date,patient_id,staff_id) values ('APT-1460',TO_DATE('19-05-2024 10:30', 'dd-MM-yyyy hh:mi'),34,347)
INTO appointment (appointment_ref,appointment_date,patient_id,staff_id) values ('APT-3086',TO_DATE('17-05-2024 10:45', 'dd-MM-yyyy hh:mi'),27,366)
INTO appointment (appointment_ref,appointment_date,patient_id,staff_id) values ('APT-2249',TO_DATE('15-04-2024 10:00', 'dd-MM-yyyy hh:mi'),61,366)
INTO appointment (appointment_ref,appointment_date,patient_id,staff_id) values ('APT-6086',TO_DATE('10-01-2024 11:00', 'dd-MM-yyyy hh:mi'),24,394)
SELECT 1 FROM DUAL
```

4 row(s) inserted.

```sql
INSERT ALL
INTO medical_record (medical_record_id,diagnosis,height_at_visit,weight_at_visit,blood_pressure,appointment_ref) VALUES ('MR-98770','Anaemia',1.77,63.91,'98/82 mmHg','APT-6086')
INTO medical_record (medical_record_id,diagnosis,height_at_visit,weight_at_visit,blood_pressure,appointment_ref) VALUES ('MR-10145','Acute chest pain',1.76,97.90,'111/71 mmHg','APT-8008')
INTO medical_record (medical_record_id,diagnosis,height_at_visit,weight_at_visit,blood_pressure,appointment_ref) VALUES ('MR-51164','Internal bleeding from accident',1.76,75.36,'130/66 mmHg','APT-3241')
INTO medical_record (medical_record_id,diagnosis,height_at_visit,weight_at_visit,blood_pressure,appointment_ref) VALUES ('MR-79038','Routine checkup all vital signs normal',1.76,87.39,'108/86 mmHg','APT-3344')
SELECT 1 FROM DUAL
```

4 row(s) inserted.

4.3 **SQL Queries for Business Analysis**

1. Retrieve a list of all patients with future appointments along with details about the staff and appointments.

```sql
SELECT first_name || ' ' || last_name as patient, TO_CHAR(appointment_date,  'dd Month yyyy') AS appointment_date,
       TO_CHAR(appointment_date, 'hh24:mi') AS appointment_time, staff_name, specialty
FROM patient JOIN appointment
    ON patient.patient_id = appointment.patient_id
    JOIN staff
    ON appointment.staff_id = staff.staff_id
    JOIN specialization
    ON staff.specialty_code = specialization.specialty_code
WHERE appointment_ref NOT IN (SELECT appointment_ref FROM medical_record)
```

PATIENT	APPOINTMENT_DATE	APPOINTMENT_TIME	STAFF_NAME	SPECIALTY
Kim Helen	17 May 2024	10:45	Fiona Naey	Doctor
Loveline Johnson	19 May 2024	10:30	Jack Jones	Doctor
Due Miller	15 April 2024	10:00	Fiona Naey	Doctor

2. Retrieve all data of patients who visited the cardiologist

```sql
SELECT appointment_date, diagnosis, staff_name, appointment.patient_id, specialty, department_name
FROM appointment JOIN medical_record
     ON appointment.appointment_ref = medical_record.appointment_ref
     LEFT JOIN staff
     ON appointment.staff_id = staff.staff_id
     LEFT JOIN department
     ON staff.department_id = department.department_id
     LEFT JOIN specialization
     ON staff.specialty_code = specialization.specialty_code
WHERE staff.specialty_code = 'SS-1' AND staff.department_id = 'Cardi-H'
```

APPOINTMENT_DATE	DIAGNOSIS	STAFF_NAME	PATIENT_ID	SPECIALTY	DEPARTMENT_NAME
22-JAN-23	Lower leg swelling possible infection	George Miller	33	Surgeon	Cardiology
19-JAN-23	Minor skin irritation topical medication prescribed	George Miller	60	Surgeon	Cardiology
09-JAN-22	Insulin resistance	George Miller	38	Surgeon	Cardiology

3. How many staff are there in each department?

```sql
SELECT department_name, COUNT(DISTINCT staff_id) AS staff_count
FROM department JOIN staff
ON department.department_id = staff.department_id
GROUP BY department_name
ORDER BY staff_count DESC
```

DEPARTMENT_NAME	STAFF_COUNT
Obstetrics & Gynecology	3
Internal Medicine	3
Pediatrics	2
Cardiology	1

5.0 **Security, Integrity, and Ethical Factors in Data Governance**

Data governance encompasses various principles, practices, and processes aimed at ensuring the security, integrity, and ethical use of data within an organization. In the contemporary era of data abundance, where large volumes of information are created and managed daily, efficient data governance plays a crucial role in managing risks and maintaining confidence among stakeholders.

This discussion reflects on the importance of security, integrity, and ethical factors in data governance.

5.0.1 **Security in Data Governance**

Security in data governance pertains to the protection of data from unauthorized access, breaches, and cyber threats, entailing the deployment of strong security protocols like encryption, access controls, and authentication mechanisms to safeguard sensitive data from malicious individuals. Security breaches can result in significant outcomes, such as monetary setbacks, harm to reputation, and legal implications. Therefore, organizations should prioritize security as a fundamental aspect of their data governance strategies (Loshin, 2014).

5.0.2 **Integrity in Data Governance**

Integrity in data governance ensures precision, quality, uniformity, and dependability of data across its entire lifespan. It involves actions to prevent data corruption, manipulation, or unauthorized alterations. Maintaining data integrity is essential for facilitating well-informed decision-making, conducting reliable analyses, and complying with regulatory requirements. Techniques such as data validation, error detection, and data quality management play a vital role in preserving data integrity within organizations (Bertino & Islam, 2002).

5.0.3 **Ethical Aspects of Data Governance**

Ethical considerations in data governance revolve around the responsible and ethical use of data, respecting individual privacy rights, and avoiding harm to stakeholders. Organizations must adhere to ethical principles such as transparency, accountability, and fairness when collecting, storing, and processing data. Additionally, adherence to pertinent regulations and standards, such as the General Data Protection Regulation (GDPR), as well as ethical directives established by professional organizations like the Association for Computing Machinery (ACM), is imperative. (Floridi et al., 2018).

5.1 **Security, Integrity, and Ethical Aspects of Data Governance (Scenario-Based)**

In the patient appointment management system, ensuring robust security measures, maintaining data integrity, and adhering to ethical guidelines are paramount.

5.1.1 **Security**: To ensure the security of the patient appointment management system, and avoid errors such as a breach of confidentiality, and user errors, the following measures were taken;

- Authorization and Authentication: The system developed has a strict adherence to self-authentication; users can only log in after verification using a registered email and login password.
- View Access: Only information relating to a particular user can be seen by the user. This was done to ensure patient data was handled with utmost security. An example of view created and access granted. Specific roles were created and access was granted accordingly. A patient_data view was created; it contains patient basic information and appointment date and access was granted to patients.

```
/* ROLES AND VIEW PRIVILEGES GRANTED*/
CREATE ROLE patients;
CREATE ROLE staff;
CREATE ROLE admin;

/*VIEWS CREATED*/
CREATE VIEW patient_data AS
SELECT patient.patient_id, first_name || ' ' || last_name as patient, address, date_of_birth, appointment_date, staff_name
FROM patient JOIN appointment
ON patient.patient_id = appointment.patient_id
JOIN staff
ON appointment.staff_id = staff.staff_id;

CREATE VIEW medical_records_data AS
SELECT patient.patient_id, first_name || ' ' || last_name as patient,  date_of_birth, diagnosis
FROM patient JOIN appointment
ON patient.patient_id = appointment.patient_id
JOIN medical_record
ON appointment.appointment_ref = medical_record.appointment_ref;

/*ACCESS GRANTED*/
GRANT SELECT ON patient_data TO patients;
GRANT SELECT ON medical_records_data TO staff;
```

- Discretionary Control: As seen above, roles are created for users, users are named and access or privileges are given based on the user clearance level. Privileges such as SELECT, GRANT, UPDATE, REFERENCES, and INSERT are granted to admins only for distributed control.

- Database Backup: The patient information management system was created and hosted using a cloud platform, to mitigate the risk associated with fire outbreaks, loss of physical storage computers or theft.

- Audit logs and trails of any activity on the database are stored and monitored for audits and scrutiny of access to the system. This checkmate access levels, and activities and ensures security.

5.1.2 **Data Integrity:** Integrity constraints were enforced during the creation of the database to ensure consistency, and maintain integrity and data quality. Integrity constraints such as the following were defined during the creation of the database system:

- Attribute constraints: The value type for each attribute was defined to ensure consistency and data entry in the system meets exact specifications and format to maintain integrity. Every attribute was defined with its attribute constraint. An example of this is appointment_date; it has an attribute constraint of TIMESTAMP (**refer to insertion script**).

```
-- Attribute constraint during table creation
appointment_date TIMESTAMP(0) NOT NULL

-- Attribute constraint application during data insertion
INTO appointment (appointment_ref,appointment_date,patient_id,staff_id) values ('APT-8577',TO_DATE('09/01/2022 10:00', 'dd-MM-yyyy hh:mi'),61,42)
```

- Entity constraint: These were enforced through primary keys and NOT NULL. To ensure the database system obeys the rule that each row of any table must have a unique and not null identifier (primary keys), PRIMARY KEYS were created for each table and specified NOT NULL. Not Null was also specified for any attribute that must not be null. Any null entry or repetition would automatically throw in an error. Example in table creation;

```
CREATE TABLE specialization (
    specialty_code  VARCHAR2(10) NOT NULL PRIMARY KEY,
    specialty       VARCHAR2(256) NOT NULL
)

Table created.
```

- Referential constraint: These were enforced through Foreign Keys to ensure that all foreign keys in a table reference a primary key in another table. Any entry of a foreign key that does not have a matching primary key would throw in an error and deletion of a primary key referenced by a foreign key in another table would also throw in an error. Example of a referential constraint defined during table creation;

```
CREATE TABLE appointment (
    appointment_ref  VARCHAR2(56) NOT NULL PRIMARY KEY,
    appointment_date TIMESTAMP(0) NOT NULL,
    patient_id       INTEGER NOT NULL,
    staff_id         INTEGER NOT NULL,
    FOREIGN KEY (patient_id) REFERENCES patient(patient_id) ON DELETE CASCADE,
    FOREIGN KEY (staff_id) REFERENCES staff(staff_id) ON DELETE CASCADE
)
```

Table created.

5.1.3 **Ethical Aspects of Data Governance**: To ensure patient data is handled with strict adherence to privacy laws and laws governing data protection, the database system adhered to ethical principles such as transparency, accountability, and fairness when collecting, storing, and processing data. This was implemented in the following ways

- Minimal data collection and transparency: Only needed and important data is collected and stored. Patients are informed that their information is collected and stored
- Patient Consent: Patient consent is obtained before data is collected.
- Internal data protection policies are put in place to ensure strict adherence to data protection laws such as General Data Protection Regulation (GDPR), Data Protection Act (DPA), and Common Law Duty of Confidentiality (CLDC).

All SQL Scripts for developing the patient information management system can be found here.

REFERENCES

Bertino, E., & Islam, N. (2002). Database Security—Concepts, Approaches, and Challenges. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 32(4), 421–426.

Connolly, T., & Begg, C. (2014). Database Systems: A Practical Approach to Design, Implementation, and Management. Pearson Education Limited.

Elmasri, R., & Navathe, S. B. (2016). Fundamentals of Database Systems. Pearson.

Floridi, L., Taddeo, M., & Turilli, M. (2018). The ethics of Information transparency. Springer.

Loshin, D. (2014). Data governance tools: Evaluation criteria and selection. Elsevie