

ỦY BAN NHÂN DÂN
THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC SÀI GÒN



BÁO CÁO MÔN HỌC
Cơ Sở Dữ Liệu Nâng Cao

Ngành: Công Nghệ Thông Tin

TÊN ĐỀ TÀI
Quarantine Camp Database

Nhóm 03
Trương Quang Hùng - 3121411081
Nguyễn Hoàng Bảo Huy - 3121411086
Huỳnh Duy Khánh - 3121411099
Nguyễn Hoàng Đăng Khoa - 3121411104
Mai Đức Kiên - 3121411111

Giảng viên : Phan Trọng Nhân

TPHCM, Ngày 6 Tháng 12 Năm 2023

Lời Nói Đầu

Trong quá trình phát triển phần mềm, việc xây dựng và quản lý cơ sở dữ liệu đóng vai trò quan trọng, ngay cả khi đó chỉ là một ứng dụng nhỏ. Cơ sở dữ liệu không chỉ giúp tổ chức thông tin một cách có tổ chức mà còn đảm bảo tính nhất quán, toàn vẹn và bảo mật dữ liệu. Việc này trở nên ngày càng quan trọng khi chúng ta cần chia sẻ thông tin với nhiều người sử dụng.

Môn học Cơ Sở Dữ Liệu Nâng Cao cung cấp kiến thức lý thuyết cần thiết để thiết kế cơ sở dữ liệu quan hệ. Từ việc xác định cấu trúc quan hệ cho đến thiết kế cấu trúc vật lý, mọi vấn đề được đề cập một cách chi tiết và có tổ chức.

Nhóm em được chọn đề tài "**QUARANTINE CAMP DATABASE**" để áp dụng những kiến thức đã học trong môn và củng cố kiến thức của nhóm. Trong quá trình nghiên cứu và thực hiện, nhóm đã nỗ lực tích hợp và áp dụng kiến thức, tuy nhiên, không tránh khỏi những khía cạnh có thể cần sự chỉnh sửa và hoàn thiện.

Nhóm trân trọng mọi ý kiến đóng góp và phản hồi từ thầy để bài làm của nhóm trở nên hoàn thiện hơn.

Em xin chân thành cảm ơn!

BẢNG PHÂN CÔNG NHÓM

Thành viên nhóm

Trương Quang Hùng - 3121411081
Nguyễn Hoàng Bảo Huy - 3121411086
Huỳnh Duy Khánh - 3121411099
Nguyễn Hoàng Đăng Khoa - 3121411104
Mai Đức Kiên - 3121411111

Bảng phân chia phần trăm công việc

Thành viên	Phần trăm công việc
Trương Quang Hùng	20%
Nguyễn Hoàng Bảo Huy	20%
Huỳnh Duy Khánh	20%
Nguyễn Hoàng Đăng Khoa	20%
Mai Đức Kiên	20%

MỤC LỤC

CHƯƠNG I: GIỚI THIỆU NỘI DUNG ĐỀ TÀI	1
1.1 Tổng quát đề tài.	1
1.2 Yêu cầu	2
CHƯƠNG II: KIẾN THỨC NỀN TẢNG.....	2
2.1 Tổng quan lý thuyết.....	2
2.1.1 Sơ lược về cách lưu trữ dữ liệu truyền thống.	2
2.1.2 Tổng quan về cơ sở dữ liệu (Database)	11
2.1.3 Tổng quan về hệ quản trị cơ sở dữ liệu (Database Management System - DBMS).....	12
2.1.4 Tổng quan về SQL (Structured Query Language).....	15
2.2 Giới thiệu sơ lược về Oracle.....	20
2.2.1 Oracle là gì?.....	20
2.2.2 Đôi nét về Oracle :.....	20
2.2.3 Kiến trúc cơ sở dữ liệu của Oracle	20
2.2.4 Ưu và nhược điểm :.....	22
2.2.5 Một số tính năng và tùy chọn của cơ sở dữ liệu Oracle:.....	22
2.2.6 Sơ lược về Oracle SQL Developer	22
CHƯƠNG III: THIẾT KẾ CƠ SỞ DỮ LIỆU VÀ MÔ HÌNH THỰC THỂ LIÊN KẾT	25
3.1 Các thực thể và thuộc tính của chúng	25
3.1.1 Các khái niệm cơ bản:.....	25
3.1.2 Thuộc tính của thực thể.....	28
3.2 Xây dựng mô hình thực thể liên kết	29
3.2.1 Các khái niệm.....	29
3.2.2 Xác định các quan hệ trong CSDL quản lý trại cách ly	29
3.2.3 Mô hình tổng quát dữ liệu thực thể liên kết.....	33
CHƯƠNG IV: CHUYỂN TỪ MÔ HÌNH THỰC THỂ LIÊN KẾT SANGMÔ HÌNH QUAN HỆ	34
4.1 Cách chuyển đổi.....	34
4.2 Xác định các bảng và thuộc tính của bảng:	35
4.3 Mô hình dữ liệu quan hệ.....	38
CHƯƠNG V: ÁP DỤNG CSDL TRÊN MÔI TRƯỜNG WEBSITE.....	39
5.1 Các ngôn ngữ sử dụng	39
5.2. Các Framework sử dụng	39
5.3. Sơ lược về website	39
5.3.1 Kịch bản:	39
5.3.2 Cách cài đặt:.....	39
5.3.3 Sơ lược về Website	40
CHƯƠNG VI: ĐÁNH GIÁ VÀ KẾT LUẬN.....	50

CHƯƠNG I: GIỚI THIỆU NỘI DUNG ĐỀ TÀI

1.1 Tổng quát đề tài.

QUARANTINE CAMP DATABASE

Due to the Covid-19 outbreak, a quarantine camp has been set up to isolate and monitor people under investigation for 21 days. Those people admitted to the quarantine camp are called “patient”. The camp stores patient information including unique number, full name, identity number, phone, gender, and address. In addition, it wants to record the patient comorbidities (e.g., cancer, chronic lung diseases, diabetes, heart conditions, immunocompromised state) because they will put a patient in a high-risk situation. In parallel, a patient needs to be tracked with his or her symptoms such as fever, dry cough, tiredness, aches and pains, sore throat, diarrhoea, conjunctivitis, headache, loss of taste or smell, a rash on skin, or discolouration of fingers or toes. Some of them may be serious like difficulty breathing or shortness of breath, chest pain or pressure, and loss of speech or movement. Unlike comorbidity, a patient symptom is different from time to time.

The camp has different types of people: managers, doctors, nurses, staffs, and volunteers. One doctor will be designated as the head of the camp. Each has its own responsibility. Besides, the camp has several buildings, each has many floors and rooms. Each room has a limited capacity. There are three types of room: normal room, emergency room, and recuperation room. When admitted by a staff, a patient is assigned into a room based on his or her current condition. Sometimes, a patient is moved from his or her room to the emergency room or the recuperation room. So, it is important to track a patient location history. The camp needs to know the admission date, from where the patient is moved to the camp, the staff information, and the testing information if any. A staff may admit many patients, and a patient is admitted by a staff.

The testing information includes those as described below:

- PCR test: the result is true (positive) or false (negative). In case it is positive, the camp wants to track the corresponding cycle threshold (ct) value.
- Quick test: the result is true (positive) or false (negative). In case it is positive, the camp wants to track the corresponding cycle threshold (ct) value.
- SPO2: which is the percent saturation of oxygen in the blood. The test measures blood oxygen levels, indicated by percentage (%).
- Respiratory rate: it is measured by how many breaths per minute.

A patient may have many testing during his or her stay. If the SPO2 is smaller than 96% and the respiratory rate is larger than 20 breaths per minute, the patient is marked “warning” and needs a healthcare action from the doctors. In case the patient has no clinical sign and the test is either negative or positive whose cycle threshold is larger than 30, he or she will be discharged from the camp. Neither of them, the patient will be tested for every 3 days by Quick test. It is important to track the discharge date for each patient.

A patient can receive treatment from at least one doctor. A doctor can treat many

patients at the same time, or sometimes, he has no patients to treat. The camp needs the details of each treatment such as: treatment period (start date and end date), result, and medications. Each patient is taken care of by a nurse; a nurse can take care of many inpatients at the same time. The information of a medication is also stored in the database. This information consists of a unique code, name of the medication, effects, price, and expiration date.

1.2 Yêu cầu

1. Design a fully labelled (E)ERD according to your business description. The diagram has to show appropriate entities (with key attributes underlined), relationships, cardinality ratios, and optional & mandatory membership classes (3 points).
2. Mapping your (E)ER diagram above to a relational database schema and identify all constraints not shown in your (E)ER diagram (1 point).
3. Build an application with the following requirements (6 points):
 - Programming environment: optional (desktop, web, or mobile application).
 - Programming language: optional.
 - Students need to prepare data and scripts for demonstration at the reporting session.

Quarantine Camp Database

1. Search patient information: Search results include the name, phone number and information about his/her comorbidities (1 point).
2. Add information for a new patient (1 point).
3. List details of all testing which belong to a patient (1 point).
4. Make a report that provides full information about the patient including demographic information, comorbidities, symptoms, testing, and treatment (1 point).
5. Proving one use-case of indexing efficiency in your scenarios (1 point)
6. Solving one use-case of database security in your scenarios (1 point)

CHƯƠNG II: KIẾN THỨC NỀN TẢNG

2.1 Tổng quan lý thuyết

2.1.1 Sơ lược về cách lưu trữ dữ liệu truyền thống.

1. Lưu trữ dữ liệu là gì?

- Các phương pháp và công nghệ tổng hợp nhằm ghi và lưu giữ thông tin số trên các phương tiện lưu trữ dựa trên điện từ, quang học hoặc silic đóng vai trò quan trọng trong việc lưu trữ thông tin số. Lưu trữ là một thành phần chính của các thiết bị số, vì người tiêu dùng và doanh nghiệp phụ thuộc vào nó để bảo tồn thông tin từ ảnh cá nhân đến thông tin quan trọng của doanh nghiệp.
- Thuật ngữ "lưu trữ" thường được sử dụng để mô tả các thiết bị và dữ liệu kết nối với máy tính thông qua các hoạt động nhập/ xuất (I/O), bao gồm ổ cứng, thiết bị flash, hệ thống băng đĩa và các loại phương tiện khác.

2. Disk Storage Devices (Thiết bị lưu trữ đĩa):

Disk storage devices là các thiết bị được sử dụng để lưu trữ dữ liệu trên đĩa, bao gồm các loại như ổ cứng (hard drive), ổ đĩa SSD (solid-state drive) và các thiết bị lưu trữ đám mây.

Ưu điểm chính: Là thiết bị lưu trữ phụ được ưa thích với khả năng lưu trữ cao và chi phí thấp.

Phương thức lưu trữ dữ liệu: Dữ liệu được lưu trữ dưới dạng khu vực từ tính trên bề mặt của các đĩa từ tính.

-Disk Pack (Bộ đĩa):

Cấu trúc: Một disk pack bao gồm nhiều đĩa từ tính được kết nối với một trục quay.

Mục tiêu: Tạo ra một không gian lưu trữ dữ liệu mở rộng và có khả năng quay để truy cập các đĩa khác nhau.

-Divided into Concentric Circular Tracks (Chia thành các đường tròn tập trung):

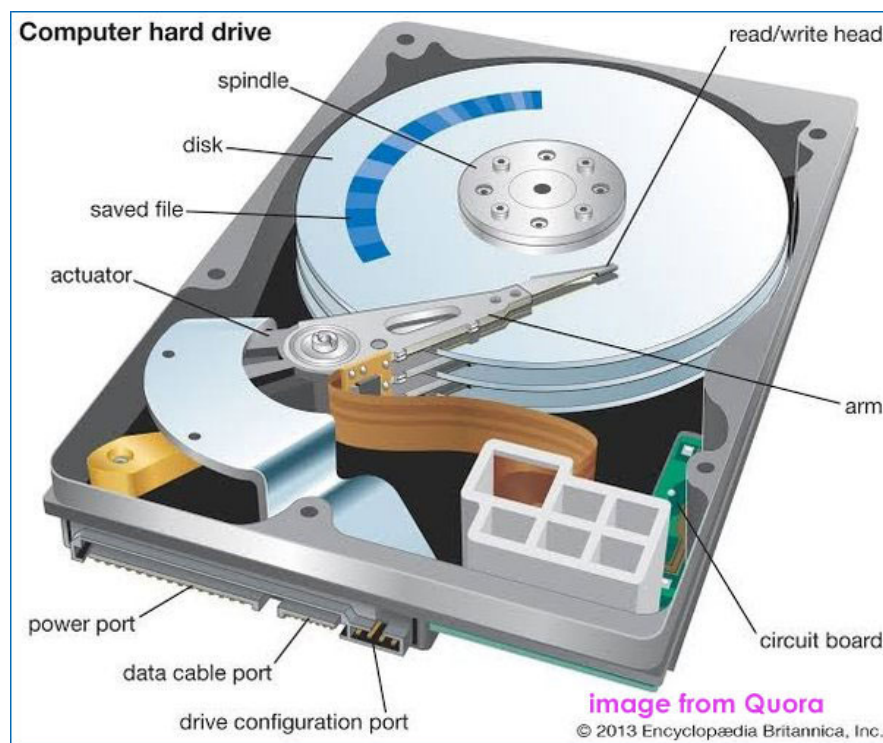
Bố cục: Các đĩa từ tính được chia thành các vòng tròn tập trung trên mỗi bề mặt đĩa.

Mục đích: Tăng cường tổ chức và quản lý dữ liệu bằng cách phân chia không gian lưu trữ thành các vùng vòng tròn.

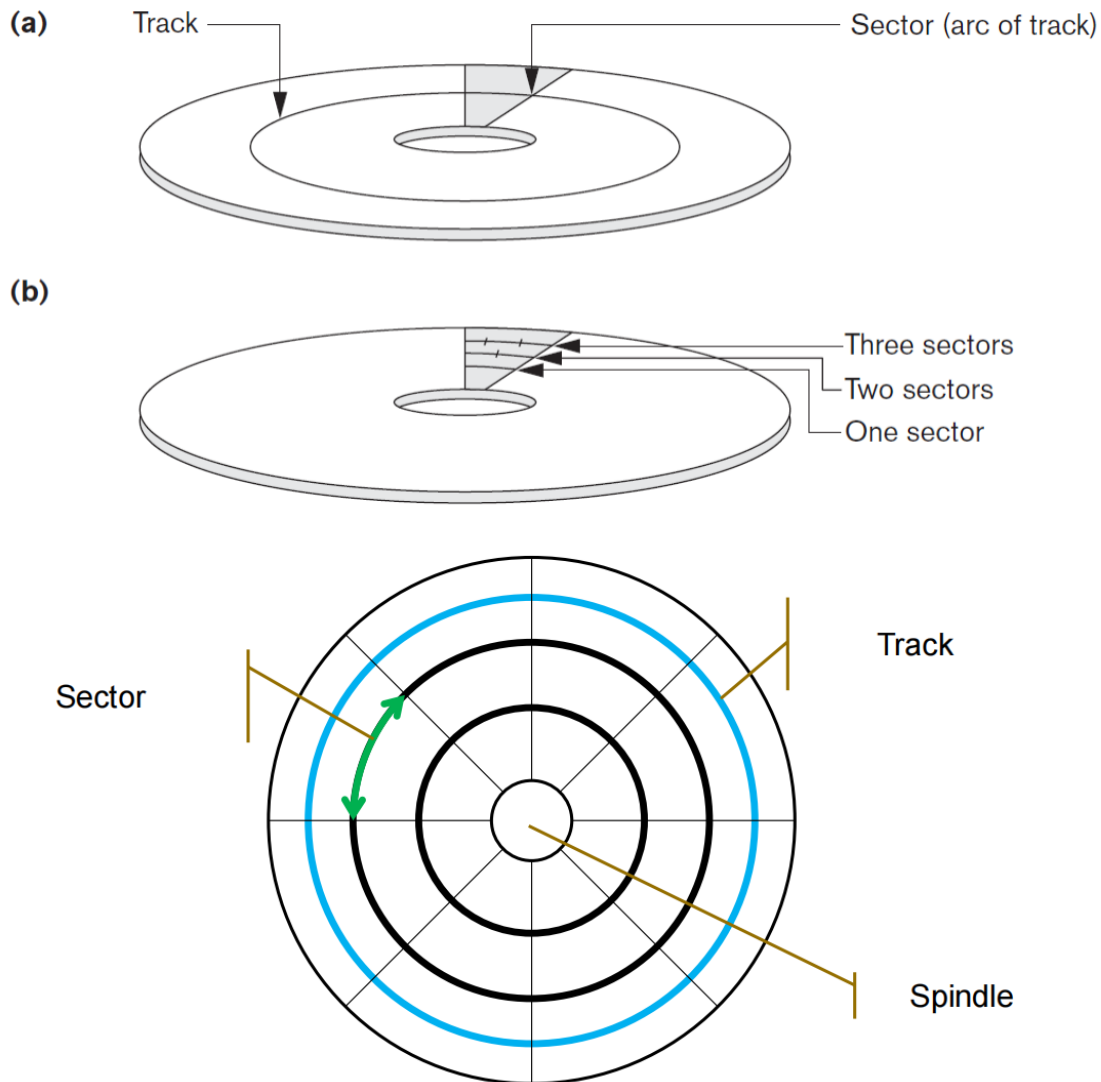
-Track Capacities (Dung lượng vòng tròn):

Biến đổi: Dung lượng của các vòng tròn thường dao động từ 4 đến 50 Kbyte.

Quản lý: Track capacities giúp xác định khả năng lưu trữ của mỗi vòng tròn và ảnh hưởng đến khả năng lưu trữ toàn bộ đĩa từ tính.



Hình 1 : Thiết bị lưu trữ đĩa từ tính.

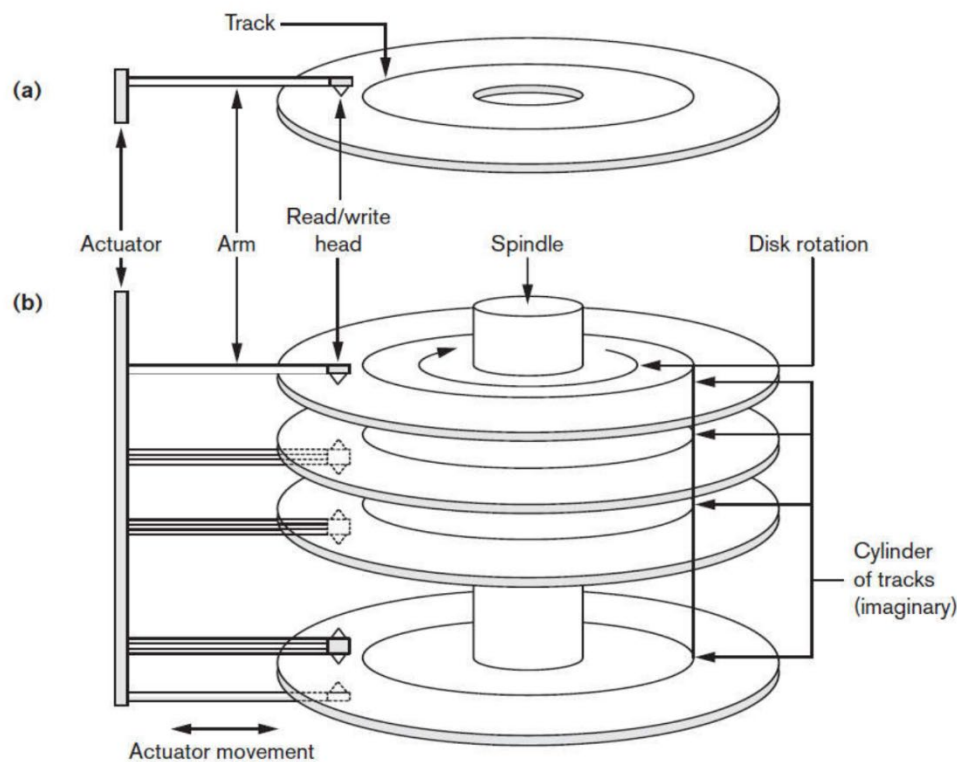


Hình 2; 3 : Hệ thống lưu trữ đĩa từ tính

- Một track được chia thành các khối hoặc sectors nhỏ.
- + Bởi vì một track thường chứa một lượng lớn thông tin.
- Một track được chia thành các khối.
- + Kích thước khối B được cố định cho mỗi hệ thống.
- + Kích thước khối điển hình dao động từ B=512 byte đến B=4096 byte.
- + Toàn bộ các khối được chuyển giữa đĩa và bộ nhớ chính để xử lý.

- Một read-write head di chuyển đến track chứa khối cần chuyển đổi.
- + quay đĩa di chuyển khối dưới read-write head để đọc hoặc ghi.

- Một địa chỉ vật lý của khối đĩa (phần cứng) bao gồm:
 - + một số xy-lanh
 - + số đường đua hoặc số bề mặt
 - + số khối
- Việc đọc hoặc ghi một khối đĩa mất thời gian do thời gian tìm kiếm s và độ trễ quay (độ trễ) rd .
- Double buffering có thể được sử dụng để tăng tốc quá trình chuyển đổi của các khối đĩa liên kề.



Hình 4: (a) Một đĩa mặt với phần cứng đọc/giữ được.
(b) Một bộ đĩa với phần cứng đọc/giữ.

Số lượng đường mạch trên một đĩa có thể dao động từ vài trăm đến vài nghìn, và dung lượng của mỗi đường mạch thường dao động từ vài chục Kbyte đến 150 Kbyte. Vì một đường mạch thường chứa một lượng lớn thông tin, nó được chia thành các khối hoặc phân khúc nhỏ hơn. Việc chia một đường mạch thành các phân khúc được lập trình sẵn trên bề mặt đĩa và không thể thay đổi được.

Việc chia một đường mạch thành các khối đĩa (hoặc trang) cùng kích thước được thiết lập bởi hệ điều hành trong quá trình định dạng đĩa (hoặc khởi tạo). Kích thước khối được cố định trong quá trình khởi tạo và không thể thay đổi động.

Kích thước khối đĩa thông thường dao động từ 512 đến 4096 byte. Các khối được tách ra bởi khoảng trống giữa các khối cố định, chứa thông tin kiểm soát được mã hóa đặc biệt trong quá trình khởi tạo đĩa.

Một đĩa là một thiết bị có thể truy cập ngẫu nhiên. Địa chỉ phần cứng của một khối bao gồm số xy-lanh, số đường mạch (số mặt trong xy-lanh trên đó đường mạch được đặt), và số khối (trong đường mạch) được cung cấp cho phần cứng đọc/ghi đĩa. Trong nhiều ổ đĩa đĩa hiện đại, một số duy nhất

được gọi là LBA (Logical Block Address), là một số giữa 0 và n (giả sử dung lượng tổng của đĩa là n+1 khối), được ánh xạ tự động đến khối phù hợp bởi bộ điều khiển ổ đĩa. Địa chỉ của một bộ đệm - một khu vực dự trữ liên tục trong bộ nhớ chính chứa một khối - cũng được cung cấp. Đối với lệnh đọc, khối từ đĩa được sao chép vào bộ đệm; trong khi đối với lệnh ghi, nội dung của bộ đệm được sao chép vào khối đĩa.

Đôi khi một số khối liên tiếp, gọi là một cụm, có thể được chuyển đổi như một đơn vị. Cơ chế phân cứng thực tế đọc hoặc ghi một khối là đầu đọc/ghi đĩa, là một phần của hệ thống được gọi là ổ đĩa. Một đĩa hoặc bộ đĩa được lắp đặt trong ổ đĩa, bao gồm một động cơ quay đĩa.

Một đầu đọc/ghi bao gồm một thành phần điện tử được gắn vào một cánh cơ khí. Bộ đĩa với nhiều bề mặt được kiểm soát bởi một số đầu đọc/ghi - một cho mỗi bề mặt (xem Hình b). Tất cả các cánh được kết nối với một bộ điều khiển gắn vào một động cơ điện khác, di chuyển các đầu đọc/ghi cùng một lúc và đặt chúng chính xác trên xi-lanh của đường mạch được chỉ định trong địa chỉ khối. Một số đơn vị đĩa có đầu đọc/ghi cố định, với số đầu bằng số đường mạch. Chúng được gọi là đĩa đầu cố định, trong khi các đơn vị đĩa với một bộ kích là đĩa đầu có thể di chuyển. Bộ điều khiển đĩa, thường được nhúng trong ổ đĩa đĩa, kiểm soát ổ đĩa và kết nối nó với hệ thống máy tính.

3. Files of Records:

Tập tin ghi chúng đề cập đến một phương pháp tổ chức và lưu trữ dữ liệu một cách có cấu trúc.

Trong ngữ cảnh này, có một số khái niệm chính được liên kết:

- Records:

Bản ghi là các đơn vị dữ liệu cá nhân chứa thông tin liên quan đến một thực thể hoặc giao dịch cụ thể. Ví dụ, trong cơ sở dữ liệu thông tin nhân viên, mỗi bản ghi có thể đại diện cho chi tiết về một nhân viên cụ thể, bao gồm tên, ID, vị trí và các thông tin khác liên quan.

- Blocking:

Chia khối ám chỉ việc nhóm nhiều bản ghi lại thành các khối hoặc khối dữ liệu. Điều này được thực hiện để cải thiện hiệu suất trong việc lưu trữ và truy xuất dữ liệu. Bằng cách lưu trữ bản ghi trong các khối, trở nên thực tế hóa hơn khi đọc hoặc ghi nhiều bản ghi cùng một lúc, giảm chi phí liên quan đến việc truy cập từng bản ghi cá nhân.

- Files of Records:

Tập tin ghi chúng đại diện cho sự tổ hợp hoặc tập hợp của các bản ghi cá nhân vào một cấu trúc tập tin. Những tập tin này có thể được tổ chức theo nhiều cách khác nhau, chẳng hạn như tuần tự hoặc ngẫu nhiên, và có thể bao gồm các kỹ thuật khác nhau như chia khối để tối ưu hóa lưu trữ và truy cập.

4. Operations on Files

Creation:

Tạo ra một tệp liên quan đến việc phân bổ không gian trong hệ thống lưu trữ và định nghĩa cấu trúc và thuộc tính của tệp.

Opening:

Mở một tệp cho phép một chương trình hoặc người dùng truy cập vào tệp để đọc, ghi hoặc thực hiện các thao tác khác. Hệ thống phải định vị tệp và chuẩn bị nó cho việc xử lý.

- Closing : Đóng một tệp là bước quan trọng sau khi thực hiện các thao tác trên đó. Nó giải phóng tài nguyên hệ thống và đảm bảo rằng các thay đổi được thực hiện trên tệp được lưu trữ một cách đúng đắn.

- Reading: Đọc liên quan đến việc truy xuất dữ liệu từ một tệp. Tùy thuộc vào cấu trúc của tệp, việc đọc có thể được thực hiện tuần tự hoặc truy xuất ngẫu nhiên.
- Writing: Ghi vào một tệp đồng nghĩa với việc thêm hoặc sửa đổi dữ liệu trong tệp. Như việc đọc, việc ghi có thể được thực hiện tuần tự hoặc truy xuất ngẫu nhiên dựa trên tổ chức của tệp.
- Updating: Cập nhật liên quan đến việc sửa đổi dữ liệu hiện tại trong một tệp. Điều này có thể bao gồm việc thay đổi giá trị, thêm bản ghi mới hoặc xóa bản ghi.
- Seeking: Seeking cho phép di chuyển con trỏ tệp đến một vị trí cụ thể trong tệp. Điều này quan trọng cho các thao tác truy xuất ngẫu nhiên.
- Deleting: Xóa một tệp liên quan đến việc loại bỏ nó khỏi hệ thống lưu trữ. Hành động này không thể đảo ngược và cần phải thận trọng để tránh mất dữ liệu.
- Truncating: Truncate một tệp liên quan đến việc loại bỏ một phần của tệp, thường từ cuối. Nó có thể được sử dụng để điều chỉnh kích thước của tệp hoặc loại bỏ dữ liệu không cần thiết.
- Renaming: Đổi tên thay đổi tên của một tệp trong khi vẫn giữ nội dung của nó nguyên vẹn. Đây là một thao tác phổ biến để tổ chức.
- Copying: Sao chép một tệp tạo ra một bản sao với cùng hoặc tên khác nhau. Điều này có thể hữu ích để sao lưu hoặc chia sẻ.
- Locking and Unlocking: Khóa một tệp ngăn chặn các quy trình khác truy cập vào nó đồng thời, đảm bảo tính nhất quán của dữ liệu. Mở khóa giải phóng khóa, cho phép các quy trình khác truy cập vào tệp.

5. Unordered Files & Ordered Files & Hashed Files

a) Unordered Files:

- Definition: Unordered files là các cấu trúc lưu trữ dữ liệu trong đó các bản ghi được đặt ngẫu nhiên hoặc không có một trình tự cụ thể. Không có trình tự hay sắp xếp nào của các bản ghi dựa trên một khóa hoặc bất kỳ tiêu chí nào được định rõ.
- Access: Việc truy xuất dữ liệu trong unordered files thường liên quan đến việc tìm kiếm tuần tự trong toàn bộ tệp. Điều này có thể dẫn đến thời gian truy xuất chậm hơn so với ordered files, đặc biệt là khi tìm kiếm một bản ghi cụ thể.
- Insertion and Deletion: Việc thêm và xóa bản ghi trong unordered files thường khá đơn giản. Tuy nhiên, các hoạt động này có thể vẫn yêu cầu tìm kiếm bản ghi cụ thể.

b) Ordered Files:

- Definition: Ordered files là các cấu trúc lưu trữ dữ liệu trong đó các bản ghi được sắp xếp theo một trình tự cụ thể dựa trên một khóa hoặc tiêu chí được quyết định trước. Khóa này thường được chọn để tối ưu hóa các hoạt động tìm kiếm và truy xuất dữ liệu.
- Access: Việc truy xuất dữ liệu trong ordered files hiệu quả hơn, đặc biệt là trong các hoạt động tìm kiếm, vì nó cho phép tìm kiếm nhị phân hoặc các thuật toán tìm kiếm tối ưu hóa khác. Sắp xếp theo trình tự giúp đơn giản hóa và tăng tốc quá trình truy xuất dữ liệu.
- Insertion and Deletion: Việc thêm và xóa bản ghi trong ordered files có thể yêu cầu duy trì thứ tự của các bản ghi, có thể bao gồm di chuyển các bản ghi hiện tại để chứa bản ghi mới hoặc đã xóa. Quá trình này có thể mất thời gian hơn so với unordered files.

c) Hashed Files:

- Definition: Hashed files sử dụng hàm băm để ánh xạ các khóa vào vị trí trong không gian lưu trữ. Ánh xạ này cho phép truy cập trực tiếp vào các bản ghi dựa trên khóa mà không cần tìm kiếm tuần tự.

- Access: Việc truy xuất dữ liệu trong hashed files thường rất nhanh chóng, vì hàm băm cho phép tính toán trực tiếp vị trí lưu trữ cho một khóa cụ thể.
- Insertion and Deletion: Việc thêm và xóa bản ghi trong hashed files có thể hiệu quả, miễn là hàm băm vẫn hiệu quả. Tuy nhiên, va chạm (hai khóa ánh xạ vào cùng một vị trí) cần được giải quyết bằng các kỹ thuật như chuỗi hoặc open addressing.

6. RAID Technology:

a) Khái niệm :

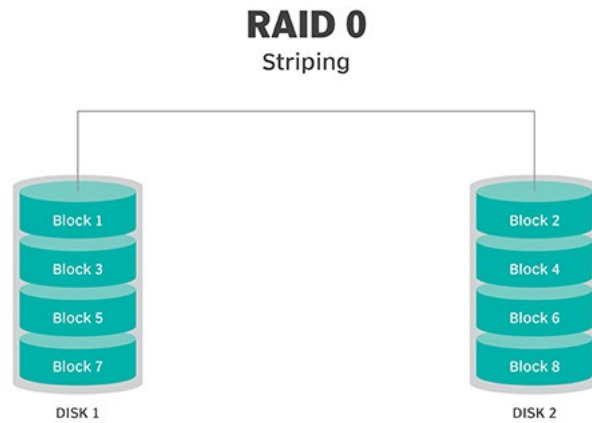
- RAID (hệ thống đĩa độc lập dự phòng) là một cách lưu trữ dữ liệu giống nhau ở các địa điểm khác nhau trên nhiều ổ đĩa cứng hoặc ổ đĩa rắn (SSDs) khác nhau để bảo vệ dữ liệu trong trường hợp ổ đĩa gặp sự cố. Tuy nhiên, có nhiều cấp độ RAID khác nhau và không phải tất cả đều có mục tiêu cung cấp tính dự phòng.



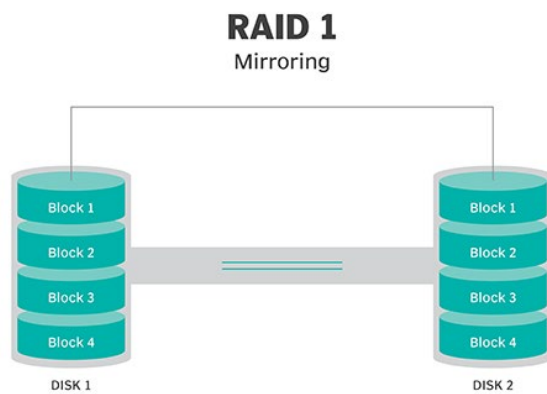
Hình 5: Một ổ đĩa cứng trong một mảng RAID.

b) RAID levels:

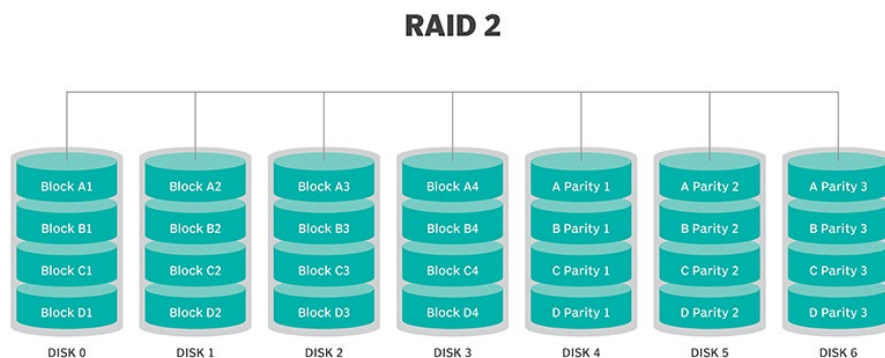
Raid level 0 : không có dữ liệu dự phòng và do đó có hiệu suất ghi tốt nhất.



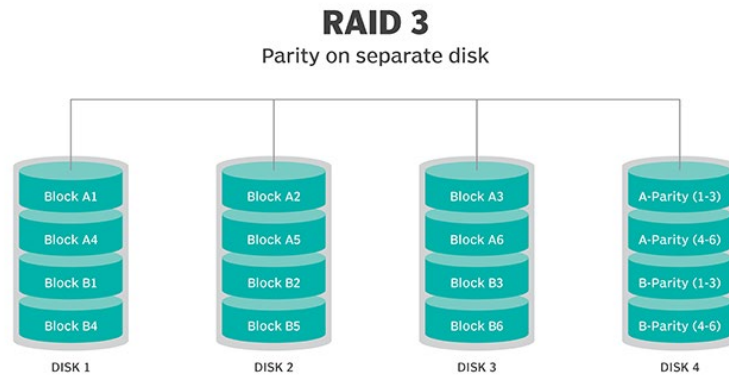
Raid level 1: sử dụng ổ đĩa gương.



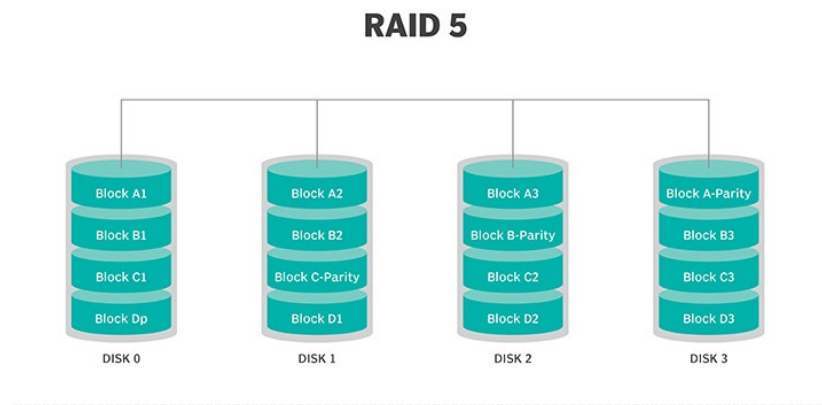
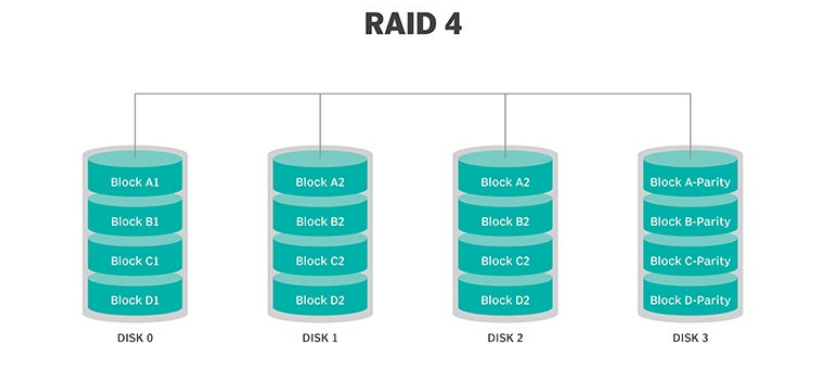
Raid level 2 : sử dụng tính dự phòng giống như bộ nhớ bằng cách sử dụng mã Hamming, chứa các bit dự phòng cho các tập hợp con trùng lặp khác nhau của các thành phần. Cấp độ 2 bao gồm cả phát hiện và sửa lỗi.



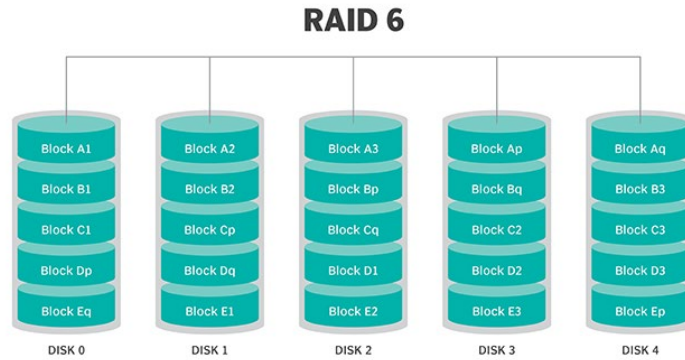
Raid level 3: sử dụng một ổ đĩa dự phòng duy nhất dựa vào bộ điều khiển ổ đĩa để xác định ổ đĩa nào đã gặp sự cố.



Raid levels 4 và 5: sử dụng data striping ở cấp độ khối, với cấp độ 5 phân phối dữ liệu và thông tin dự phòng qua tất cả các ổ đĩa.



Raid level 6: áp dụng kế hoạch dự phòng P + Q sử dụng mã Reed-Soloman để bảo vệ chống đến hai sự cố ổ đĩa bằng cách sử dụng chỉ hai ổ đĩa dự phòng.



2.1.2 Tổng quan về cơ sở dữ liệu (Database)

1. Database :

Cơ sở dữ liệu là một bộ sưu tập được tổ chức của dữ liệu có liên quan. Thuật ngữ "tổ chức" có nghĩa là dữ liệu được lưu trữ sao cho người sử dụng có thể lưu trữ, thao tác và truy xuất dữ liệu một cách dễ dàng. Thuật ngữ "liên quan" có nghĩa là cơ sở dữ liệu thường được tạo ra để lưu trữ dữ liệu về một chủ đề cụ thể.

Ví dụ, nếu bạn tạo một cơ sở dữ liệu cho sinh viên, nó sẽ chứa dữ liệu về sinh viên như số thứ tự, tên, địa chỉ và các thông tin khác về sinh viên. Tương tự, nếu cơ sở dữ liệu liên quan đến nhân viên của một tổ chức, nó sẽ chứa dữ liệu về nhân viên như mã nhân viên, cấp bậc và lương, v.v. Tất cả dữ liệu trong cơ sở dữ liệu được sắp xếp trong các bảng.

2. Table:

Bảng là đối tượng cơ bản của cấu trúc cơ sở dữ liệu. Mục đích cơ bản của một bảng là lưu trữ dữ liệu. Nó bao gồm các dòng và cột. Bảng là một cách rất thuận tiện để lưu trữ dữ liệu. Người sử dụng có thể dễ dàng thao tác dữ liệu trong một bảng.

Serial No	Name	Roll no	Total marks	Marks obtained
1	Usman	554480	1100	850
2	Abdullah	554481	1100	796
3	Ejaz	554482	1100	725

3. Row/ Record:

Hàng là phần ngang của bảng. Đó là một bộ sưu tập của các trường liên quan. Ví dụ, có ba hàng trong bảng ở trên. Mỗi hàng chứa một bản ghi về người khác nhau.

1	Usman	554480	1100	850
---	-------	--------	------	-----

4. Column/ Field:

Cột là phần đứng của bảng.

Ví dụ, tất cả các giá trị dưới trường Tên trong bảng ở trên tạo thành một cột.

Name
Usman
Abdullah
Ejaz

VÍ DỤ VỀ CƠ SỞ DỮ LIỆU:

5. DANH BẠ ĐIỆN THOẠI

Danh bạ điện thoại là một ví dụ đơn giản về một cơ sở dữ liệu. Một danh bạ điện thoại lưu trữ các số điện thoại của các người khác nhau. Bạn có thể dễ dàng tìm kiếm bất kỳ số điện thoại nào từ danh bạ điện thoại vì tất cả các số điện thoại được lưu trữ theo cách tổ chức.

6. THƯ VIỆN

Một thư viện chứa hàng nghìn sách. Việc quản lý hồ sơ của tất cả những cuốn sách này mà không có một cơ sở dữ liệu sẽ rất khó khăn. Hệ thống cơ sở dữ liệu có thể được sử dụng để lưu trữ hồ sơ về sách, thành viên của thư viện, việc cấp và thu hồi sách, v.v. Người sử dụng có thể sử dụng cơ sở dữ liệu để dễ dàng tìm kiếm sách cần thiết. Cơ sở dữ liệu này có thể hỗ trợ rất nhiều cho công việc nghiên cứu.

7. TÀI KHOẢN

Một cơ sở dữ liệu được sử dụng để kiểm soát hệ thống tài khoản của một tổ chức. Cơ sở dữ liệu tài khoản lưu trữ hồ sơ về tất cả các giao dịch tài chính. Nó có thể dễ dàng thực hiện các phép tính khác nhau để tìm thông tin về doanh nghiệp như lợi nhuận hàng năm, cân đối kế toán và sổ cái v.v.

8. TRƯỜNG ĐẠI HỌC

Một trường đại học có nhiều sinh viên ở các lớp khác nhau. Một cơ sở dữ liệu có thể được sử dụng để lưu trữ hồ sơ của sinh viên, giao dịch học phí, kỳ thi, thông tin và các dữ liệu khác về trường đại học. Bạn cũng có thể lưu trữ điểm danh của sinh viên trong cơ sở dữ liệu.

2.1.3 Tổng quan về hệ quản trị cơ sở dữ liệu (Database Management System - DBMS)

1. Database Management System:

"Một hệ thống quản lý cơ sở dữ liệu (DBMS) là phần mềm hệ thống để tạo ra và quản lý cơ sở dữ liệu. Hệ thống DBMS cung cấp người sử dụng và lập trình viên một cách hệ thống để tạo ra, truy xuất, cập nhật và quản lý dữ liệu."

- DBMS (Database Management System) là bộ sưu tập các phần mềm được sử dụng để tổ chức thông tin trong một cơ sở dữ liệu, có thể chứa các quy trình đầu vào dữ liệu, xác minh, lưu trữ, truy xuất và kết hợp.

- DBMS là các ứng dụng tương tác với người sử dụng và các ứng dụng khác để thu thập và phân tích dữ liệu.

- DBMS là một bộ phần mềm máy tính cung cấp giao diện giữa người sử dụng và cơ sở dữ liệu.

- Chúng liên quan chặt chẽ đến thuật ngữ "cơ sở dữ liệu", và khi sử dụng thông thường, thuật ngữ "cơ sở dữ liệu" thường đề cập đến cả DBMS và dữ liệu mà nó thao tác.

2. Mục Tiêu của Hệ Thống Quản Lý Cơ Sở Dữ Liệu (DBMS):

- Đây là những mục tiêu quan trọng của DBMS.

a) KHẢ NĂNG CHIA SẺ:

Khả năng chia sẻ tài nguyên dữ liệu là một mục tiêu cơ bản của quản lý cơ sở dữ liệu. Trong nghĩa đầy đủ nhất, điều này có nghĩa là nhiều người và các quy trình khác nhau sử dụng cùng một dữ liệu thực tế gần như đồng thời.

b) KHẢ NĂNG TRUY CẬP:

Khả năng truy cập có nghĩa là đưa dữ liệu của một tổ chức đến người sử dụng dữ liệu đó. Hệ thống quản lý tài nguyên dữ liệu nên dễ dàng tiếp cận đối với những người trong tổ chức - đưa ra dữ liệu khi và nơi nó cần thiết, và theo cách và hình thức nó cần.

c) TÍNH TOÀN VỆ:

Tính toàn vẹn có nghĩa là dữ liệu của bạn là chính xác và nhất quán. DBMS có nhiều kiểm tra hợp lệ giúp dữ liệu của bạn hoàn toàn chính xác và nhất quán.

d) KHẢ NĂNG PHÁT TRIỂN:

Khả năng phát triển liên quan đến khả năng của DBMS thay đổi để đáp ứng nhu cầu người dùng ngày càng tăng và công nghệ tiến bộ. Khả năng phát triển là đặc điểm của hệ thống cải thiện sự sẵn có trong tương lai của tài nguyên dữ liệu. Khả năng phát triển bao gồm sự mở rộng hoặc thu nhỏ, cả hai điều này có thể xảy ra khi hệ thống thay đổi để phù hợp với nhu cầu và mong muốn thay đổi liên tục của môi trường sử dụng.

3. Các thành phần của hệ thống quản lý cơ sở dữ liệu (DBMS):

Hệ thống quản lý cơ sở dữ liệu (DBMS) bao gồm nhiều thành phần. Mỗi thành phần đóng vai trò rất quan trọng trong môi trường hệ thống quản lý cơ sở dữ liệu. Các thành phần chính của hệ thống quản lý cơ sở dữ liệu bao gồm:

A) DỮ LIỆU:

Dữ liệu là thành phần quan trọng nhất của DBMS. Mục tiêu chính của DBMS là xử lý dữ liệu. Trong DBMS, cơ sở dữ liệu được định nghĩa, xây dựng, và sau đó dữ liệu được lưu trữ, cập nhật và truy xuất từ cơ sở dữ liệu.

A) PHẦN MỀM:

Phần mềm là thành phần chủ yếu của một DBMS. Đó là bộ chương trình được sử dụng để xử lý cơ sở dữ liệu và kiểm soát cũng như quản lý toàn bộ cơ sở dữ liệu máy tính hóa.

- Phần mềm DBMS chính là thành phần phần mềm quan trọng nhất trong hệ thống tổng thể.

- Hệ điều hành kèm theo phần mềm mạng được sử dụng trong mạng, để chia sẻ dữ liệu của cơ sở dữ liệu giữa nhiều người sử dụng.

- Các chương trình ứng dụng được phát triển bằng các ngôn ngữ lập trình như C++, Visual Basic được sử dụng để truy cập cơ sở dữ liệu trong hệ thống quản lý cơ sở dữ liệu. Mỗi chương trình chứa các lệnh yêu cầu DBMS thực hiện các thao tác trên cơ sở dữ liệu, bao gồm truy xuất, cập nhật, xóa dữ liệu, vv. Chương trình ứng dụng có thể là truyền thống hoặc trực tuyến trên các máy trạm hoặc thiết bị đầu cuối.

PHẦN CỨNG:

Phần cứng bao gồm một bộ thiết bị điện tử vật lý như máy tính (kèm theo các thiết bị nhập/ra như ổ đĩa cứng), thiết bị lưu trữ, kênh nhập/ra, các thiết bị điện cơ học tạo ra giao diện giữa máy tính và các hệ thống thế giới thực, và còn nhiều hơn nữa. Không thể triển khai DBMS mà không có các thiết bị phần cứng, trong mạng, máy tính mạnh với tốc độ xử lý dữ liệu cao và một thiết bị lưu trữ với dung lượng lưu trữ lớn là cần thiết như một máy chủ cơ sở dữ liệu.

B) NHÂN SỰ:

Những người liên quan đến hệ thống cơ sở dữ liệu được gọi là nhân sự. Các loại người trong hệ thống cơ sở dữ liệu bao gồm:

- Người lập trình ứng dụng.
- Người quản trị cơ sở dữ liệu.
- Người sử dụng cuối.

*** NGƯỜI LẬP TRÌNH ỨNG DỤNG:**

Những người viết chương trình ứng dụng bằng các ngôn ngữ lập trình (như Visual Basic, Java hoặc C++) để tương tác với cơ sở dữ liệu được gọi là Người Lập Trình Ứng Dụng.

*** NGƯỜI QUẢN TRỊ CƠ SỞ DỮ LIỆU:**

Người chịu trách nhiệm quản lý toàn bộ hệ thống quản lý cơ sở dữ liệu được gọi là quản trị cơ sở dữ liệu hoặc đơn giản là DBA.

*** NGƯỜI SỬ DỤNG CUỐI:**

Người sử dụng cuối là những người tương tác với hệ thống quản lý cơ sở dữ liệu để thực hiện các hoạt động khác nhau trên cơ sở dữ liệu như truy xuất, cập nhật, chèn, xóa dữ liệu, vv.

C) THỦ TỤC:

Đây là các hướng dẫn và quy tắc hỗ trợ cách sử dụng DBMS, và trong thiết kế và vận hành cơ sở dữ liệu, sử dụng các thủ tục được tài liệu hóa để hướng dẫn người sử dụng vận hành và quản lý nó.

4. Đặc điểm của hệ thống quản lý cơ sở dữ liệu (DBMS):

Có nhiều công cụ quản lý cơ sở dữ liệu (DBMS) cung cấp các tính năng và chức năng khác nhau. Dưới đây là những đặc điểm có thể và cần có của một DBMS tiêu biểu:

A) CẤU TRÚC DỮ LIỆU:

Tất cả thông tin trong một kho dữ liệu số được tổ chức thành một cấu trúc phân cấp rõ ràng với các bản ghi, bảng hoặc đối tượng. Mọi thông tin bạn thêm vào cơ sở dữ liệu của mình sẽ được tổ chức trong một loại catalogue nào đó, giúp bạn dễ dàng tìm kiếm và cập nhật bản ghi của mình sau này.

B) TÙY CHỈNH CƠ SỞ DỮ LIỆU:

Ngoài các phần tử mặc định và cần thiết (bản ghi, bảng hoặc đối tượng) tạo nên cấu trúc của một cơ sở dữ liệu, có thể tạo các phần tử tùy chỉnh để đáp ứng nhu cầu cụ thể của người sử dụng.

C) TRUY XUẤT DỮ LIỆU:

Hệ thống quản lý cơ sở dữ liệu (DBMS) chấp nhận đầu vào dữ liệu từ người sử dụng và lưu trữ

nó. Người sử dụng sau đó có thể truy cập cơ sở dữ liệu để lấy lại bản ghi của họ dưới dạng tệp, in ấn hoặc xem trên màn hình.

D) NGÔN NGỮ TRUY VẤN:

Một DBMS tiêu biểu cho phép sử dụng các ngôn ngữ truy vấn để thu thập, tìm kiếm, sắp xếp, sửa đổi và các hoạt động khác giúp người sử dụng thao tác các bản ghi của họ trong cơ sở dữ liệu.

E) TRUY CẬP ĐA NGƯỜI DÙNG:

DBMS cung cấp truy cập đa người dùng đến tất cả các loại thông tin được lưu trữ trong cùng một kho dữ liệu. Cũng có một tính năng bảo mật ngăn chặn một số người sử dụng xem và/hoặc sửa đổi một số loại dữ liệu cụ thể.

F) DỮ LIỆU MÔ TẢ (METADATA):

Phần mềm quản lý cơ sở dữ liệu của DBMS cung cấp một thư viện dữ liệu mô tả (hoặc từ điển dữ liệu) mô tả cách cơ sở dữ liệu được tổ chức và các phần tử (đối tượng, tệp liên quan, bản ghi, vv.) tạo nên cấu trúc của nó.

2.1.4 Tổng quan về SQL (Structured Query Language)

1. Structured Query Language (SQL):

- SQL là một ngôn ngữ cơ sở dữ liệu tiêu chuẩn được sử dụng để tạo, duy trì và truy xuất cơ sở dữ liệu quan hệ.

*Cơ sở dữ liệu quan hệ là gì?

Cơ sở dữ liệu quan hệ có nghĩa là dữ liệu được lưu trữ và truy xuất dưới dạng các mối quan hệ (bảng).

2. Lịch sử của SQL:

- Phiên bản ban đầu của SQL, được gọi là SEQUEL (Structured English Query Language), được thiết kế bởi một trung tâm nghiên cứu của IBM vào năm 1974 và 1975.

- SQL lần đầu tiên được giới thiệu như một hệ thống cơ sở dữ liệu thương mại vào năm 1979 bởi Oracle Corporation.

- SQL trở thành một tiêu chuẩn của Viện Tiêu chuẩn Quốc gia Hoa Kỳ (ANSI) vào năm 1986 và của Tổ chức Tiêu chuẩn Hóa Quốc tế (ISO) vào năm 1987.

3. Những lý do chọn SQL :

- SQL có cú pháp đơn giản và dễ học.

- Cung cấp khả năng truy cập dữ liệu trong các hệ thống quản lý cơ sở dữ liệu quan hệ.

- Hỗ trợ người dùng mô tả cấu trúc dữ liệu.

- Cho phép định nghĩa dữ liệu trong cơ sở dữ liệu và thực hiện các thao tác trên dữ liệu đó.

- Dễ nhúng vào các ngôn ngữ khác thông qua việc sử dụng mô-đun, thư viện và trình biên dịch SQL.

- Hỗ trợ người dùng tạo và xóa cơ sở dữ liệu cũng như bảng.

- Cho phép quản lý quyền truy cập đối với các bảng và dữ liệu.

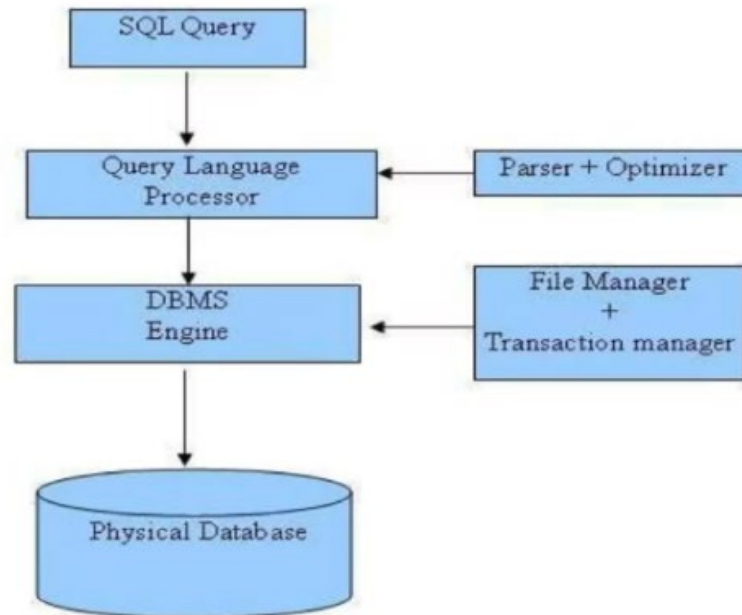
4. Quy trình của SQL:

- Khi bạn thực hiện một lệnh SQL cho bất kỳ Hệ quản trị cơ sở dữ liệu quan hệ (RDBMS) nào, hệ thống xác định cách thực hiện yêu cầu của bạn và trình độ SQL tìm cách hiệu nghiệm vụ đó.

- Có nhiều thành phần được bao gồm trong quy trình này. Các thành phần này bao gồm Trình điều phối truy vấn, Các động cơ Tối ưu hóa, Động cơ Truy vấn Cổ điển và Động cơ Truy vấn SQL,

v.v.

- Động cơ truy vấn cổ điển xử lý tất cả các truy vấn không phải SQL, nhưng động cơ truy vấn SQL sẽ không xử lý các tệp logic.



Hình 6 : Bểu đồ hoạt động của SQL.

5. DDL (Data Definition Language):

- Được sử dụng để xác định cấu trúc hoặc hệ thống cơ sở dữ liệu. DDL cũng được sử dụng để chỉ định các thuộc tính bổ sung của dữ liệu.

- Điều này bao gồm các thay đổi về cấu trúc của bảng như việc tạo bảng hoặc cơ sở dữ liệu, thay đổi bảng, xóa bảng, làm trống bảng hoặc đổi tên bảng, v.v.

Command	Description
CREATE	To create database and its objects like (table, index, view, store procedure, function, and trigger
ALTER	Alters the structure of existing database
DROP	Delete objects from database
TRUNCATE	Remove all records from a table including all spaces allocated for the records are removed
COMMENT	Add comments to the data dictionary
RENAME	Rename an object

6. DML (Data Manipulation Language):

- DML xử lý thao tác dữ liệu và bao gồm các câu lệnh SQL phổ biến như SELECT, INSERT,

UPDATE, DELETE. Nó được sử dụng để lưu trữ, sửa đổi, truy xuất, xóa và cập nhật dữ liệu trong cơ sở dữ liệu.

Command	Description
SELECT	Retrieves certain records from one or more tables
INSERT	Creates a record
UPDATE	Modifies records
DELETE	Deletes records

7. DCL (Data Control Language):

Bao gồm các lệnh để cấp quyền và thu quyền từ người dùng cơ sở dữ liệu.

Command	Description
GRANT	Gives a privilege to user
REVOKE	Takes back privileges granted from user

8. TCL (Transaction Control Language):

- TCL (Transaction Control Language) xử lý Giao dịch trong cơ sở dữ liệu

Command	Description
COMMIT	Commits a transaction
ROLLBACK	Rollback a transaction in case of any error occurs
SAVEPOINT	To rollback the transaction make in points within groups
SET TRANSACTION	Specify characteristics for the transaction

PROCEDURE :

- Một chương trình con là một đơn vị/chương trình mô-đun thực hiện một nhiệm vụ cụ thể. Những chương trình con này được kết hợp để tạo ra các chương trình lớn hơn. Điều này thường được gọi là "Modular design."

- Một chương trình con có thể được gọi bởi một chương trình con hoặc chương trình khác, được gọi là chương trình gọi.

PL/SQL cung cấp hai loại chương trình con:

- Functions (Hàm): Những chương trình con này trả về một giá trị duy nhất, thường được sử dụng để tính toán và trả về một giá trị.

- Procedures (Thủ tục): Những chương trình con này không trực tiếp trả về giá trị, thường được sử dụng để thực hiện một hành động.

* Creating a Procedure :

```
CREATE [OR REPLACE] PROCEDURE procedure_name
[(parameter_name [IN | OUT | IN OUT] type [, ...])] {IS |
AS}
BEGIN
    < procedure_body >
END procedure_name;
```

9. Function:

- Một hàm PL/SQL giống như một thủ tục ngoại trừ việc nó trả về một giá trị.

```

CREATE [OR REPLACE] FUNCTION function_name
[(parameter_name [IN | OUT | IN OUT]
  datatype )]
RETURN datatype
{IS | AS}
BEGIN
  function_body
END function_name;

```

10. Cursor :

- Là một khu vực làm việc tạm thời được tạo trong bộ nhớ hệ thống khi một câu lệnh SQL được thực thi.

%FOUND	Returns TRUE if the last fetch returned a row, or FALSE if the last fetch failed to return a row.
%NOTFOUND	The logical opposite of %FOUND.
%ROWCOUNT	Before the first fetch, returns 0. When a cursor is opened, %ROWCOUNT is zeroed. Thereafter, returns the number of rows fetched so far. The number is incremented if the latest fetch returned a row.
%ISOPEN	If a cursor is open, returns TRUE; otherwise, it returns FALSE.

Bảng : Lấy thông tin trạng thái về một con trỏ

11. Trigger :

- Các Trigger là các chương trình lưu trữ, được tự động thực thi hoặc kích hoạt khi xảy ra một số sự kiện. Thực tế, các Trigger được viết để thực thi khi có bất kỳ sự kiện sau đây xảy ra:

- + Một câu lệnh manipulation cơ sở dữ liệu (DML) (DELETE, INSERT hoặc UPDATE).
- + Một câu lệnh định nghĩa cơ sở dữ liệu (DDL) (CREATE, ALTER hoặc DROP).
- + Một hoạt động cơ sở dữ liệu (SERVERERROR, LOGON, LOGOFF, STARTUP hoặc SHUTDOWN).

-Các Trigger có thể được định nghĩa trên bảng, view, schema hoặc cơ sở dữ liệu mà sự kiện được liên kết.

***Creating Trigger:**

```

CREATE [OR REPLACE] TRIGGER schema.trigger_name
BEFORE | AFTER | INSTEAD OF
DELETE | INSERT | UPDATE [OF columns list] [OR ...]
ON schema.table_name
[REFERENCING OLD [AS] <old_name> | NEW [AS]
  <new_name>]
[FOR EACH ROW]
[WHEN (condition)]
BEGIN
  PL/SQL_block | call_procedure_statement;
END trigger_name;

```

2.2 Giới thiệu sơ lược về Oracle

2.2.1 Oracle là gì?

Oracle là một công ty toàn cầu chuyên cung cấp các sản phẩm và dịch vụ trong lĩnh vực công nghệ thông tin. Nổi tiếng nhất là hệ thống quản lý cơ sở dữ liệu quan hệ (RDBMS) của họ, có tên chính thức là Oracle Database. Oracle Database là một hệ thống quản lý cơ sở dữ liệu mạnh mẽ và được sử dụng rộng rãi trên toàn thế giới.

Oracle cung cấp nhiều sản phẩm và dịch vụ khác ngoài cơ sở dữ liệu, bao gồm ứng dụng kinh doanh, hệ thống máy tính, phần mềm phân tích dữ liệu, công cụ phát triển, và nhiều giải pháp công nghệ khác. Công ty này cũng có xu hướng mở rộng sang lĩnh vực dịch vụ điện toán đám mây.

Ngoài ra, "Oracle" cũng thường được sử dụng để chỉ hệ thống quản lý cơ sở dữ liệu của công ty, cũng như các sản phẩm và dịch vụ liên quan đến nó. Oracle Database được biết đến với hiệu suất cao, khả năng mở rộng và tính bảo mật, là sự lựa chọn phổ biến cho nhiều doanh nghiệp và tổ chức.

2.2.2 Đôi nét về Oracle :

- Oracle Corporation đã đi một quãng đường dài để trở thành tập đoàn công nghệ đa quốc gia như ngày nay. Lịch sử phong phú của nó theo dõi sự tiến triển qua quá trình phát triển và tiếp thị các hệ thống phần cứng máy tính, phần mềm doanh nghiệp và chính công nghệ cơ sở dữ liệu nổi tiếng thế giới là Oracle.

- Thành lập của Oracle:

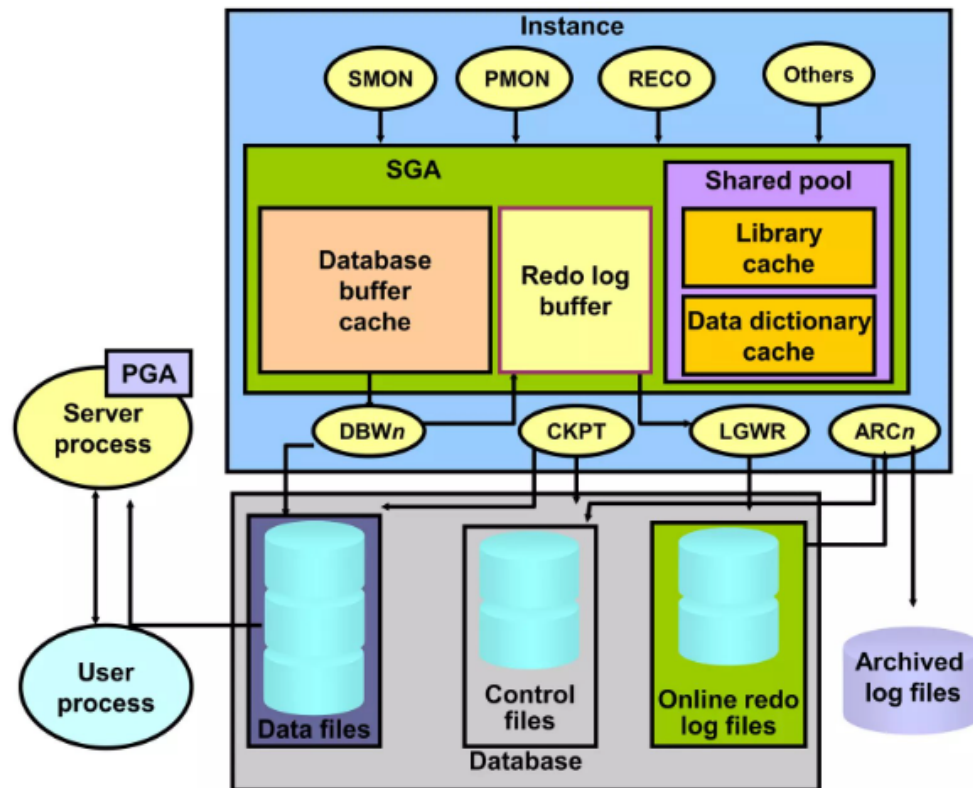
Năm 1977, Larry Ellison, Bob Miner và Ed Oates bắt đầu công ty tư vấn Software Development Laboratories, sau đó trở thành Relational Software, Inc. (RSI).

Năm 1983, RSI chuyển đổi thành Oracle Systems Corporation và sau đó trở thành Oracle Corporation.

- Chấp nhận chiến lược internet:

Năm 1995, Oracle trở thành một trong những công ty phần mềm lớn đầu tiên thông báo về chiến lược internet của mình.

2.2.3 Kiến trúc cơ sở dữ liệu của Oracle



- Database Instance (Phiên bản Cơ sở dữ liệu):

+SGA (System Global Area): Là một vùng bộ nhớ chung được chia sẻ giữa tất cả các tiến trình của một phiên bản cơ sở dữ liệu. Bao gồm nhiều thành phần như Buffer Cache (Bộ nhớ đệm), Shared Pool (Bể chia sẻ), và Large Pool (Bể lớn).

+Background Processes (Tiến trình nền): Đảm nhiệm các nhiệm vụ quản lý hệ thống như quản lý bộ nhớ, quản lý quy trình, và giám sát.

+PGA (Process Global Area): Là vùng bộ nhớ được cấp phát cho mỗi tiến trình cá nhân, chứa dữ liệu và biến môi trường của tiến trình.

- Database Files (Tập Cơ sở dữ liệu):

+ Datafiles (Tập dữ liệu): Là nơi lưu trữ dữ liệu thực sự của cơ sở dữ liệu, bao gồm các bảng, chỉ mục và dữ liệu hệ thống.

+ Control Files (Tập kiểm soát): Chứa thông tin quan trọng về cơ sở dữ liệu như tên cơ sở dữ liệu, đường dẫn của các tập dữ liệu, và trạng thái của cơ sở dữ liệu.

+ Redo Log Files (Tập ghi lại lại): Ghi lại các thay đổi dữ liệu trong cơ sở dữ liệu, giúp đảm bảo tính nhất quán và khả năng khôi phục.

- Memory Structures (Các cấu trúc bộ nhớ):

+ Buffer Cache: Lưu trữ bản sao của dữ liệu được đọc từ tập dữ liệu, giúp giảm thời gian truy xuất dữ liệu từ đĩa.

+ Shared Pool: Chứa các câu lệnh SQL đã được thực thi, lên lịch thực thi và các đối tượng dữ liệu chia sẻ khác.

+ Java Pool: Lưu trữ các đối tượng Java và mã máy ảnh của chúng.

+ Large Pool: Được sử dụng để lưu trữ bộ nhớ đệm cho các tiến trình lớn và các thực thể khác.

- Process Architecture (Kiến trúc Tiến trình):

- + Server Processes (Tiến trình máy chủ): Xử lý các yêu cầu của người dùng, thực hiện các truy vấn SQL và gửi kết quả về máy khách.
- + Background Processes (Tiến trình nền): Thực hiện các nhiệm vụ quản lý hệ thống như quản lý bộ nhớ, quản lý quy trình và sao lưu.
- + Client Processes (Tiến trình máy khách): Là các tiến trình chạy trên máy người dùng, kết nối với máy chủ cơ sở dữ liệu thông qua mạng để gửi yêu cầu và nhận kết quả.

- Logical Structures (Các cấu trúc logic):

- + Tablespaces (Không gian bảng): Phân chia cơ sở dữ liệu thành các đơn vị lưu trữ dữ liệu logic.
- + Schema (Khung cảnh): Chứa các đối tượng như bảng, chỉ mục, và các đối tượng khác.
- + Segments (Đoạn): Là phần vùng của một bảng hoặc chỉ mục trong không gian bảng.
- + Extent (Phạm vi): Là một phần của một đoạn, là nơi dữ liệu thực sự được lưu trữ.

2.2.4 Ưu và nhược điểm :

- Oracle Database được biết đến với nhiều ưu điểm nổi bật. Đầu tiên, hiệu suất cao của nó là điểm mạnh, đặc biệt là trong việc xử lý các cơ sở dữ liệu lớn. Khả năng mở rộng của Oracle cũng là một lợi thế quan trọng, cho phép doanh nghiệp linh hoạt mở rộng cơ sở dữ liệu theo nhu cầu tăng cao. Tính đáng tin cậy cao của Oracle được thiết lập thông qua sự sẵn sàng cao, đảm bảo hoạt động của cơ sở dữ liệu luôn ổn định và tin cậy. Cơ chế phục hồi và sao lưu mạnh mẽ giúp đảm bảo an toàn dữ liệu và khả năng khôi phục nhanh chóng khi có sự cố. Cuối cùng, với thương hiệu nổi tiếng trên thị trường, Oracle trở thành một lựa chọn đáng tin cậy và được ưa chuộng rộng rãi trong hệ thống quản lý cơ sở dữ liệu.

- Tuy nhiên, không phải mọi hệ thống hoàn hảo, và Oracle cũng có nhược điểm của mình. Độ phức tạp trong quản lý là một thách thức, đặc biệt đối với người dùng không quen với sự phức tạp của nó. Chi phí là một điểm đáng lưu ý, có thể cao hơn so với một số hệ thống quản lý cơ sở dữ liệu khác. Quản lý và bảo trì hiệu quả đòi hỏi kỹ năng chuyên môn chuyên sâu. Mặc dù Oracle hoạt động trên nhiều hệ điều hành, nhưng sự tương thích với các ngôn ngữ chuyên dụng chỉ hỗ trợ một số hệ điều hành cụ thể. Do đó, việc chọn lựa Oracle cần xem xét cẩn thận để đảm bảo rằng các ưu và nhược điểm được cân nhắc một cách toàn diện trong quá trình triển khai và sử dụng.

2.2.5 Một số tính năng và tùy chọn của cơ sở dữ liệu Oracle:

Một số dịch vụ và lựa chọn của cơ sở dữ liệu Oracle dựa trên thông tin trên bao gồm:

- Upgrade Service: Dịch vụ nâng cấp giúp cải thiện và đưa cơ sở dữ liệu lên phiên bản mới nhất, đồng thời đảm bảo tính tương thích và hiệu suất tối ưu.
- Migration Services: Dịch vụ di chuyển hỗ trợ chuyển đổi dữ liệu từ môi trường cơ sở dữ liệu hiện tại sang cơ sở dữ liệu Oracle, đảm bảo sự chuyển đổi suôn sẻ và an toàn.
- Disaster Recovery Services: Dịch vụ khôi phục thảm họa tạo ra một hệ thống sao lưu để bảo vệ cơ sở dữ liệu khỏi mất mát dữ liệu và đảm bảo khả năng khôi phục sau sự cố.
- Database Management Services: Dịch vụ quản lý cơ sở dữ liệu cung cấp các công cụ và tiện ích để theo dõi, duy trì và tối ưu hóa hiệu suất của cơ sở dữ liệu Oracle.

2.2.6 Sơ lược về Oracle SQL Developer

a) Oracle SQL Developer:

-Oracle SQL developer là giao diện người dùng (Graphical user interface), cho phép người dùng và quản trị viên hệ cơ sở dữ liệu thực hiện các tác vụ liên quan tới hệ cơ sở dữ liệu. Là một công cụ giúp tăng năng suất, SQL Developer giúp người dùng tiết kiệm tối đa thời gian và tối đa hoá khả năng làm việc.

-Oracle SQL Developer là một môi trường phát triển tích hợp miễn phí giúp đơn giản hóa quá trình phát triển và quản lý cơ sở dữ liệu Oracle trong cả môi trường truyền thống và triển khai trên đám mây. SQL Developer cung cấp toàn bộ quá trình phát triển ứng dụng PL/SQL của bạn từ đầu đến cuối, một bảng công việc để chạy truy vấn và scripts, một bảng điều khiển DBA để quản lý cơ sở dữ liệu, một giao diện báo cáo, một giải pháp mô hình dữ liệu hoàn chỉnh, và một nền tảng di động để chuyển đổi cơ sở dữ liệu của bên thứ ba của bạn sang Oracle.

b) Các tính năng chính :

- Chạy Truy Vấn và Scripts:

+ SQL Developer cung cấp một bảng worksheet cho phép bạn chạy các truy vấn SQL và scripts một cách thuận tiện.

+ Dễ dàng thực hiện các tác vụ truy vấn cơ bản và phức tạp trên cơ sở dữ liệu Oracle của bạn.

- Phát Triển Ứng Dụng PL/SQL:

+ SQL Developer là một Integrated Development Environment (IDE) đầy đủ tính năng cho phát triển ứng dụng PL/SQL.

+ Hỗ trợ việc viết, sửa đổi và kiểm thử mã PL/SQL.

-Quản Lý Cơ Sở Dữ Liệu:

+Bảng điều khiển DBA cung cấp các công cụ quản lý cơ sở dữ liệu để theo dõi và điều khiển các khía cạnh quan trọng của hệ thống.

-Mô Hình Hóa Dữ Liệu:

+ SQL Developer cung cấp một giải pháp mô hình hóa dữ liệu, giúp bạn thiết kế và quản lý cấu trúc cơ sở dữ liệu một cách trực quan.

- Di Động Cơ Sở Dữ Liệu và Chuyển Đổi:

+ Hỗ trợ di động cơ sở dữ liệu từ các hệ thống quản lý cơ sở dữ liệu của bên thứ ba sang Oracle.

+ Cung cấp các công cụ chuyển đổi để di chuyển dữ liệu từ Oracle On-Premises sang Oracle Cloud.

- Hỗ Trợ Môi Trường Đám Mây:

+ SQL Developer có khả năng hỗ trợ việc chuyển đổi từ Oracle On-Premises sang Oracle Cloud, đồng thời cung cấp các tính năng liên quan đến môi trường đám mây.

c) Destop:

Trên máy tính, Oracle SQL Developer cung cấp một bộ tính năng đầy đủ:

- Full PL/SQL IDE:

+ SQL Developer hoạt động như một môi trường phát triển tích hợp đầy đủ cho PL/SQL.

+ Hỗ trợ các nhiệm vụ như biên dịch, gỡ lỗi, tải liệu hóa, kiểm thử, và kiểm soát mã nguồn cho các ứng dụng PL/SQL.

-Tương Tác với Cơ Sở Dữ Liệu:

+Người dùng có thể nhấp chuột, duyệt và quản lý nội dung của cơ sở dữ liệu Oracle của họ trực tiếp từ giao diện máy tính.

+Điều này giúp dễ dàng khám phá và thao tác các đối tượng cơ sở dữ liệu.

-Quản Lý Cơ Sở Dữ Liệu:

+ SQL Developer hỗ trợ quản lý nhiều khía cạnh của Cơ sở Dữ liệu Oracle.

+ Người dùng có thể quản lý hiệu suất cơ sở dữ liệu, bảo mật, lưu trữ, cài đặt và nhiều tính năng khác, tất cả đều từ môi trường máy tính.

-Phát Triển Dịch Vụ RESTful:

+ Công cụ này hỗ trợ phát triển các dịch vụ RESTful, cho phép người dùng tạo và quản lý các API.

+ Có tích hợp với Oracle REST Services, giúp tương tác mượt mà với các dịch vụ RESTful.

-Giải Pháp Mô Hình Hóa Dữ Liệu Hoàn Chỉnh:

+SQL Developer cung cấp một giải pháp mô hình hóa dữ liệu toàn diện, bao gồm sơ đồ quan hệ thực thể (ERDs), báo cáo, scripts SQL và Data Definition Language (DDL), kiểm soát mã nguồn, và khả năng so sánh các phiên bản khác nhau (DIFFs) của mô hình dữ liệu.

d) Command line:

Trên dòng lệnh, Oracle SQL Developer cung cấp một loạt các tính năng:

-Chỉnh Sửa Hiện Đại Trực Tiếp cho SQL và PL/SQL:

+Cung cấp khả năng chỉnh sửa hiện đại trực tiếp cho các câu lệnh SQL và PL/SQL và các đoạn mã.

-Hoàn Thành Tự Động cho Các Từ Khóa, Lệnh và Tên Đối Tượng Oracle:

+Hỗ trợ hoàn thành tự động cho các từ khóa, lệnh và tên đối tượng trong Oracle, giúp tiết kiệm thời gian và giảm lỗi nhập liệu.

-Lịch Sử SQL - Truy Cập Tất Cả Các Truy Vấn và Đoạn Mã Bạn Đã Thực Hiện:

+Cho phép truy cập vào lịch sử của tất cả các truy vấn và đoạn mã bạn đã chạy trước đó.

- Tự Động Định Dạng Kết Quả Truy Vấn - Hiển Thị Kết Quả SQL Dễ Đọc hoặc Định Dạng thành JSON, XML, CSV, INSERTs, và HTML:

+ Tự động định dạng kết quả truy vấn để dễ đọc hoặc chuyển đổi thành các định dạng như JSON, XML, CSV, INSERTs, và HTML.

-Tạo DDL Cho Đối Tượng hoặc Lấy Dữ Liệu Về Đối Tượng bằng Các Lệnh Đơn Giản như DDL và INFO:

+ Cung cấp khả năng tạo mã DDL cho đối tượng hoặc truy cập thông tin về đối tượng thông qua các lệnh đơn giản như DDL và INFO.

- Tích Hợp Liquibase để Quản Lý Phiên Bản Nguồn của Oracle Dễ Dàng:

+ Hỗ trợ tích hợp với Liquibase để quản lý phiên bản nguồn của Oracle một cách thuận tiện.

- Rất Nhẹ, Chỉ Có Dung Lượng Tải và Cài Đặt 25MB:

+SQL Developer rất nhẹ, chỉ có dung lượng tải và cài đặt khoảng 25MB, giúp giữ cho ứng dụng nhẹ nhàng và chiếm ít không gian trên hệ thống.

e) Web:

- Chạy bất kỳ SQL hoặc PL/SQL nào trong Bảng công việc SQL đầy đủ chức năng (Lịch sử SQL, thông tin chi tiết, kế hoạch giải thích, và nhiều tính năng khác).

- Tạo hoặc chỉnh sửa các đối tượng schema Oracle yêu thích của bạn với các bảo mật sử dụng các hướng dẫn dễ sử dụng.

- Tạo ra hoặc chỉnh sửa biểu đồ mối quan hệ thực thể (ERDs) chi tiết hiển thị thiết kế Schema Oracle hiện tại của bạn.

- Quản lý hiệu suất và cấu hình cơ sở dữ liệu chung.

- Tạo và chỉnh sửa người dùng.

- Tải dữ liệu từ các tệp CSV, JSON và AVRO vào các bảng Oracle mới hoặc đã tồn tại.

CHƯƠNG III: THIẾT KẾ CƠ SỞ DỮ LIỆU VÀ MÔ HÌNH THỰC THỂ LIÊN KẾT

3.1 Các thực thể và thuộc tính của chúng

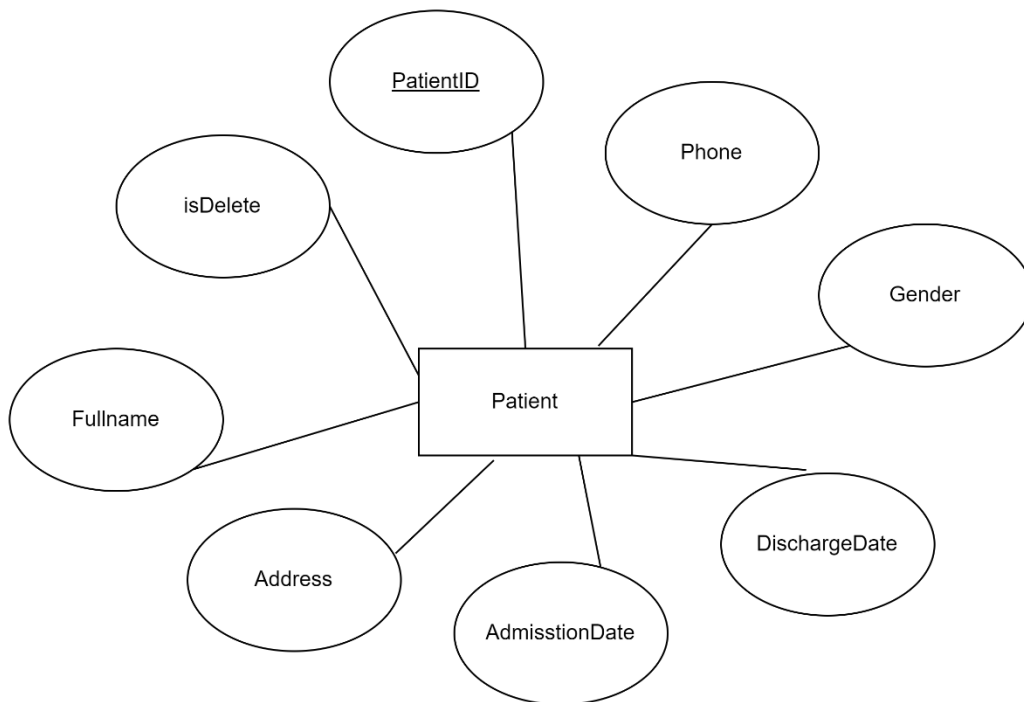
3.1.1 Các khái niệm cơ bản:

-Thực thể là khái niệm để chỉ một lớp các đối tượng có cùng đặc tính chung mà người ta muốn quản lý thông tin về nó.

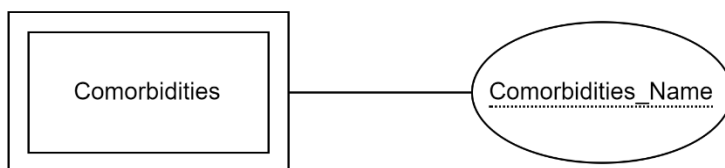
-Ở đây bao gồm: Patient, Comorbidities, Symptom, Patient_Location_History, Testing_information, Room, People, Medication, Treatment.

Một đối tượng cụ thể trong thực thể được gọi là một cá thể (còn gọi là một thể hiện của thực thể).

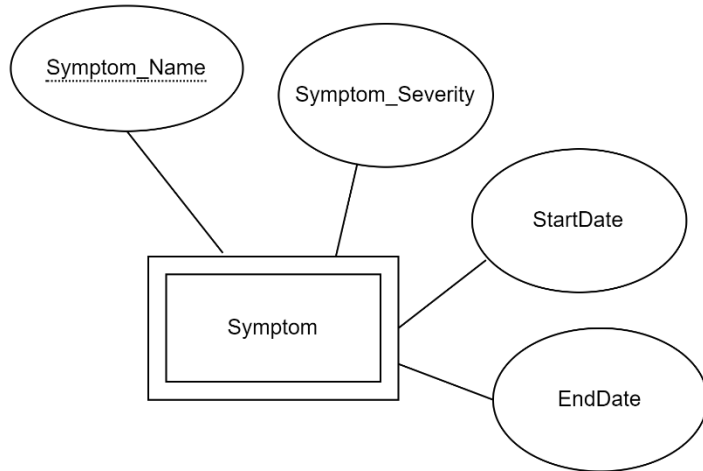
-Thực thể Patient gồm: PatientID, Phone, Gender, DischargeDate, AdmisstionDate, Address, Fullname, isDelete.



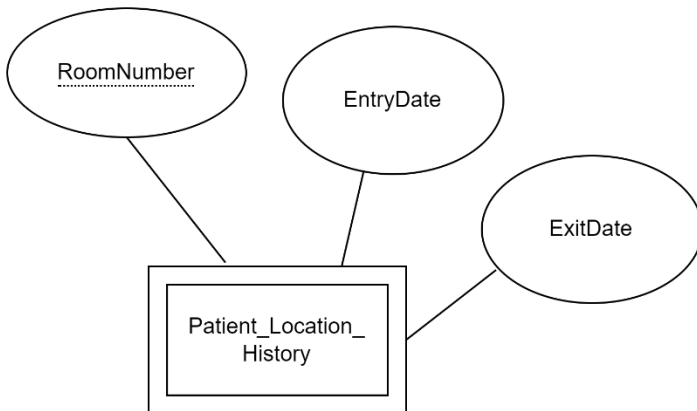
-Thực thể Comorbidities gồm:Comorbidities_Name



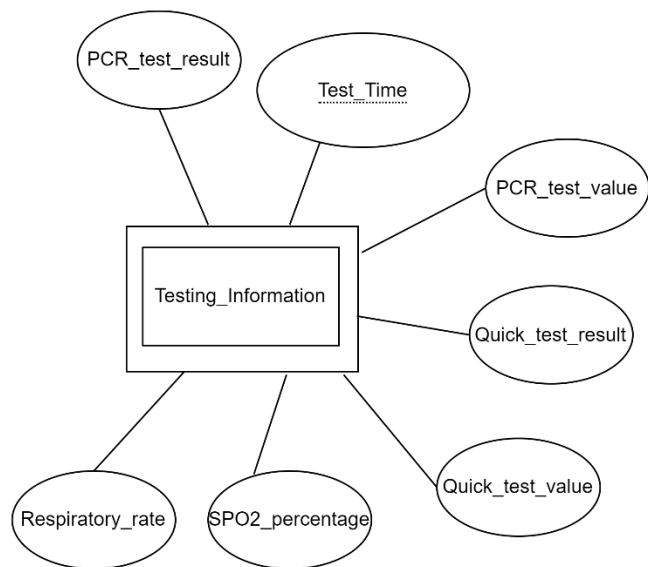
-Thực thể Symptom gồm: Symptom_Name, Symptom_Severity, StarDate, EndDate.



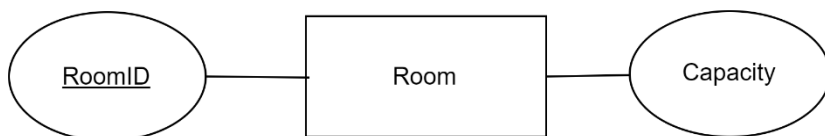
-Thực thể Patient_Location_History gồm: RoomNumber, EntryDate, ExitDate



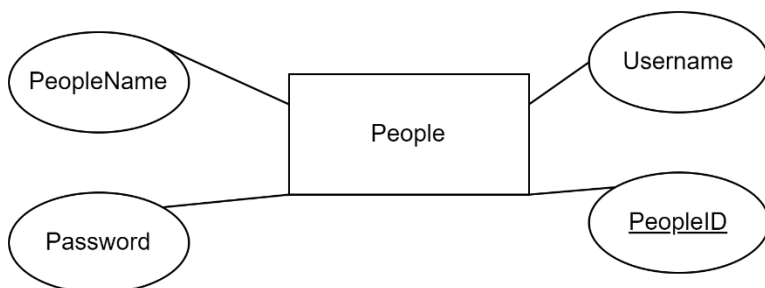
-Thực thể Testing_information: Test_Time, PCR_test_value, PCR_test_result, Quick_test_result, Quick_test_value, SPO2_percentage, Respiratory_rate



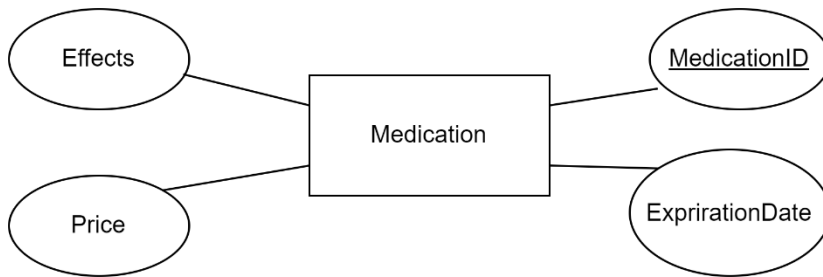
-Thực thể Room gồm:RoomID, Capacity



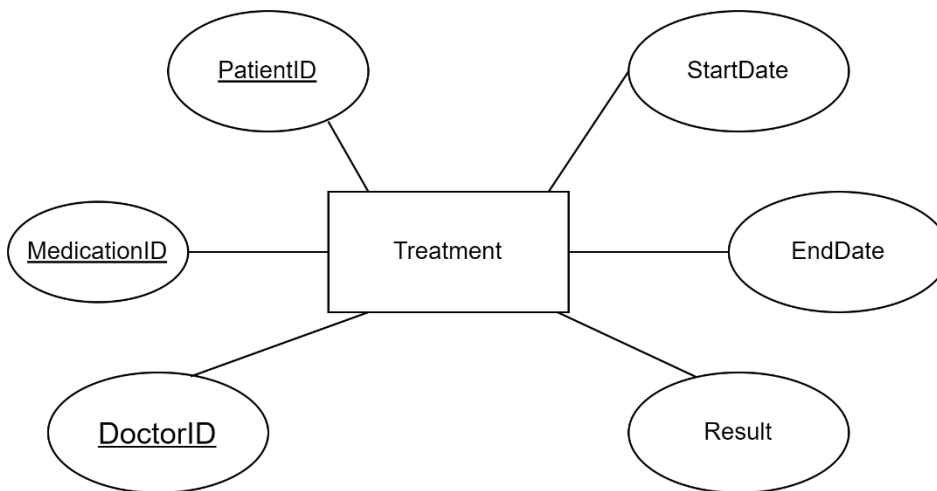
-Thực thể People gồm: PeopleID, Username, PeopleName,Password



-Thực thể Medication gồm:MedicationID, Effects, Price, ExpirationDate



-Thực thể Treatment gồm: PatientID, Medication, DoctorID, StarDate, EndDate, Result



3.1.2 Thuộc tính của thực thể

Để mô tả thông tin về một thực thể người ta thường dựa vào các đặc trưng riêng của thực thể đó. Các đặc trưng đó được gọi là thuộc tính của thực thể. Ví dụ thực thể Dược phẩm có các thuộc tính như MedicationID, Effects, Price, ExpirationDate.

Thuộc tính của thực thể bao gồm các loại sau:

- Thuộc tính định danh (còn gọi là định danh thực thể, đôi khi còn gọi là thuộc tính khoá): Là một hoặc một số thuộc tính mà giá trị của nó cho phép phân biệt các thực thể khác nhau. Một thực thể bao giờ cũng được xác định một thuộc tính định danh làm cơ sở để phân biệt các thể hiện cụ thể của nó.

Ví dụ: IdKH, IdSanPham, IdNV,...

- Thuộc tính mô tả: Là các thuộc tính mà giá trị của chúng chỉ có tính mô tả cho thực thể hay liên kết mà thôi. Hầu hết các thuộc tính trong một kiểu thực thể đều là mô tả.
- Thuộc tính tên gọi: là thuộc tính mô tả để chỉ tên các đối tượng thuộc thực thể.

Thuộc tính tên gọi để phân biệt các thực thể (tách các thực thể).

- Thuộc tính kết nối (thuộc tính khoá ngoài): là thuộc tính chỉ ra mối quan hệ giữa một thực thể đã có và một thực thể trong bảng khác. Thuộc tính kết nối giống thuộc tính mô tả thông thường trong thực thể chứa nó nhưng nó lại là thuộc tính khoá của một thực thể trong bảng khác.

3.2 Xây dựng mô hình thực thể liên kết

3.2.1 Các khái niệm

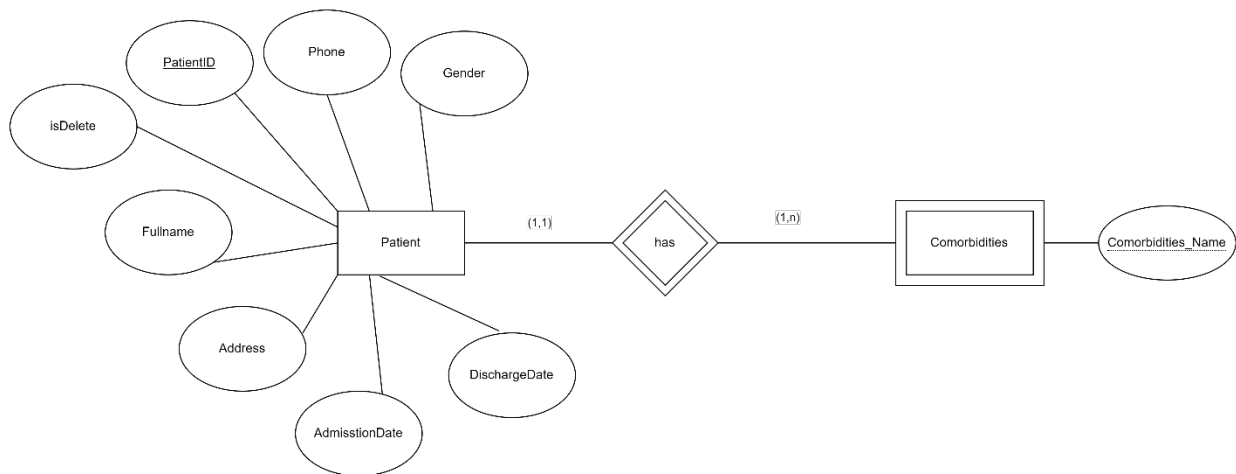
Mô hình thực thể liên kết còn được gọi là Mô Hình ER.

Mô hình ER được dùng để xây dựng mô hình hóa dữ liệu ý niệm (conceptual data modeling) nhằm biểu diễn cấu trúc và các ràng buộc của CSDL.

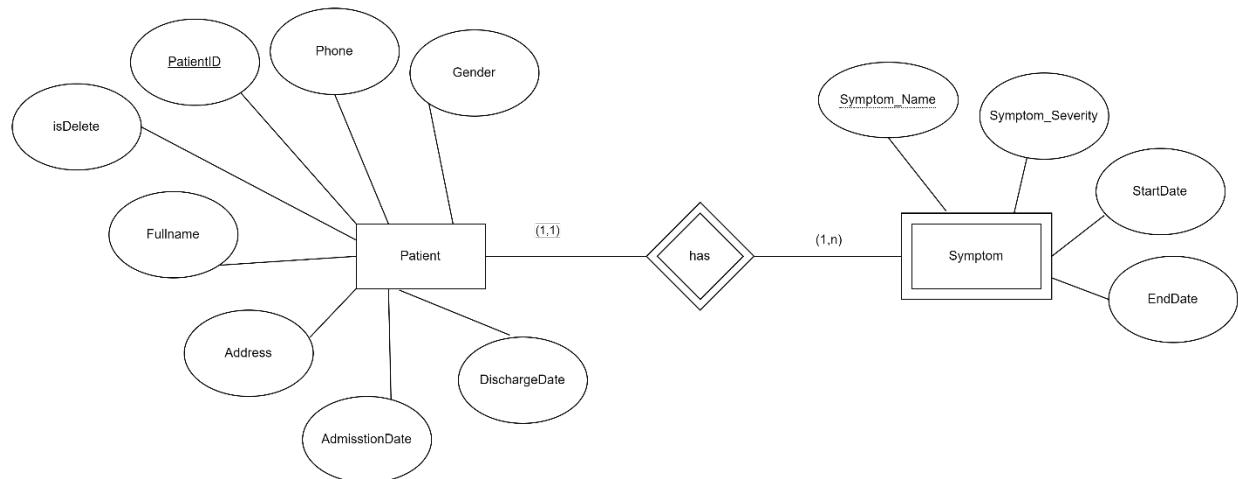
Mô hình ER như một công cụ giao tiếp giữa người thiết kế CSDL và người sử dụng cuối cùng để xây dựng CSDL trong giai đoạn phân tích. Nó độc lập với DBMS và quá trình thi công CSDL.

3.2.2 Xác định các quan hệ trong CSDL quản lý trại cách ly

- The camp stores patient information including unique number, full name, identity number, phone, gender, and address. In addition, it wants to record the patient comorbidities (e.g., cancer, chronic lung diseases, diabetes, heart conditions, immunocompromised state)

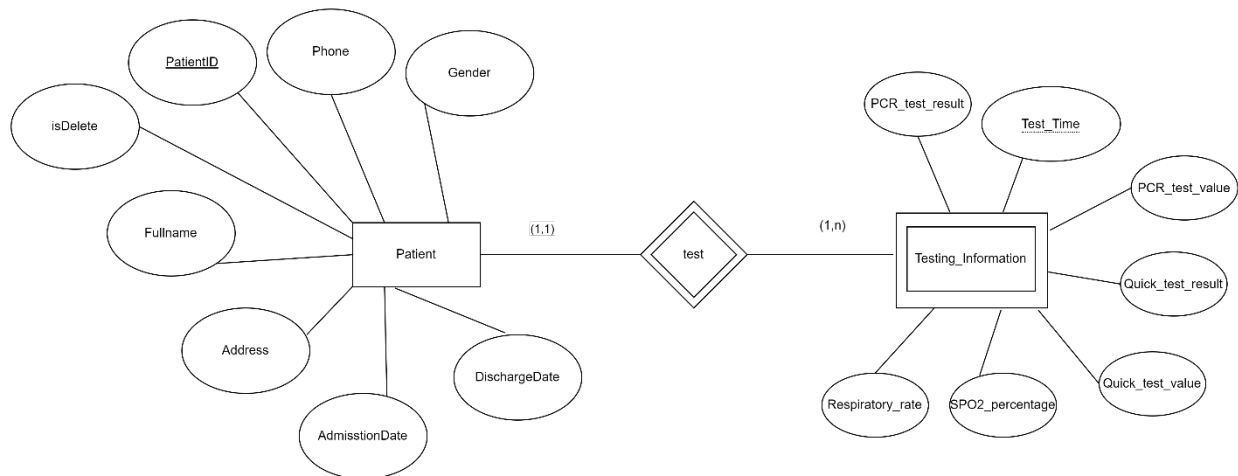


- In parallel, a patient needs to be tracked with his or her symptoms such as fever, dry cough, tiredness, aches and pains, sore throat, diarrhoea, conjunctivitis, headache, loss of taste or smell, a rash on skin, or discolouration of fingers or toes. Some of them may be serious like difficulty breathing or shortness of breath, chest pain or pressure, and loss of speech or movement.

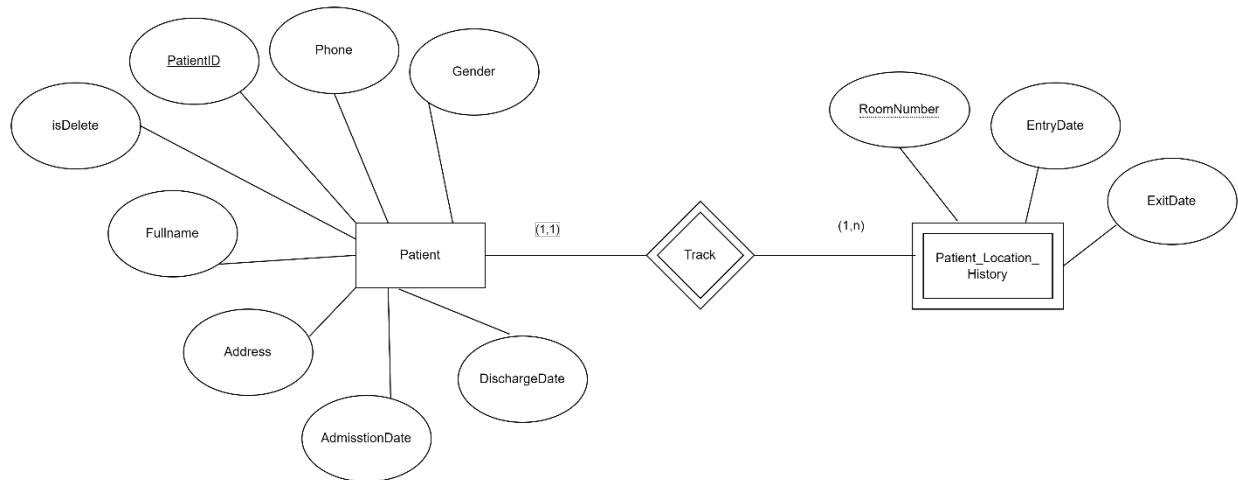


- The testing information includes those as described below:
 - PCR test: the result is true (positive) or false (negative). In case it is positive, the camp wants to track the corresponding cycle threshold (ct) value.
 - Quick test: the result is true (positive) or false (negative). In case it is positive, the camp wants to track the corresponding cycle threshold (ct) value.
 - SPO2: which is the percent saturation of oxygen in the blood. The test measures blood oxygen levels, indicated by percentage (%).
 - Respiratory rate: it is measured by how many breaths per minute.

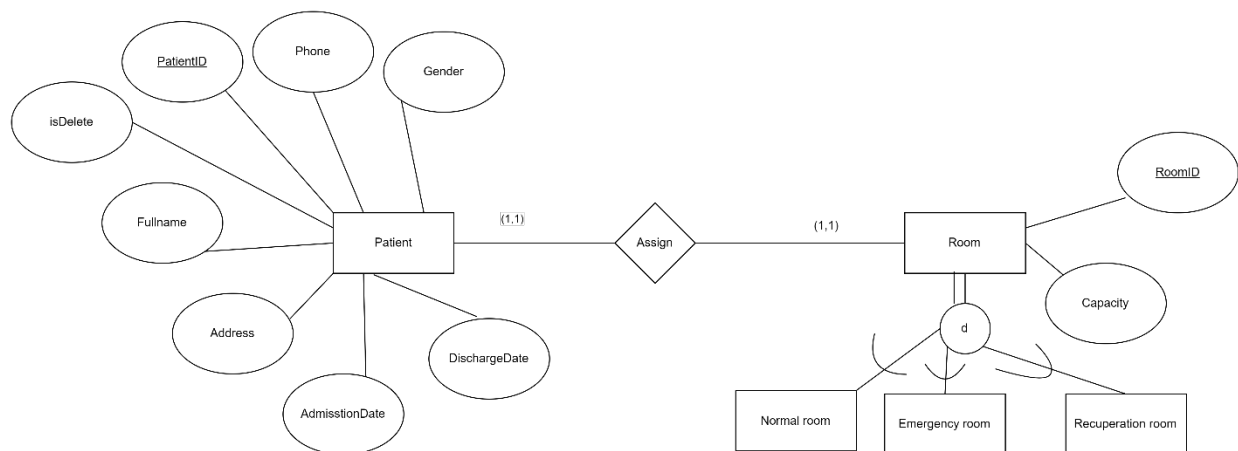
A patient may have many testing during his or her stay. If the SPO2 is smaller than 96% and the respiratory rate is larger than 20 breaths per minute, the patient is marked “warning” and needs a healthcare action from the doctors. In case the patient has no clinical sign and the test is either negative or positive whose cycle threshold is larger than 30, he or she will be discharged from the camp. Neither of them, the patient will be tested for every 3 days by Quick test. It is important to track the discharge date for each patient.



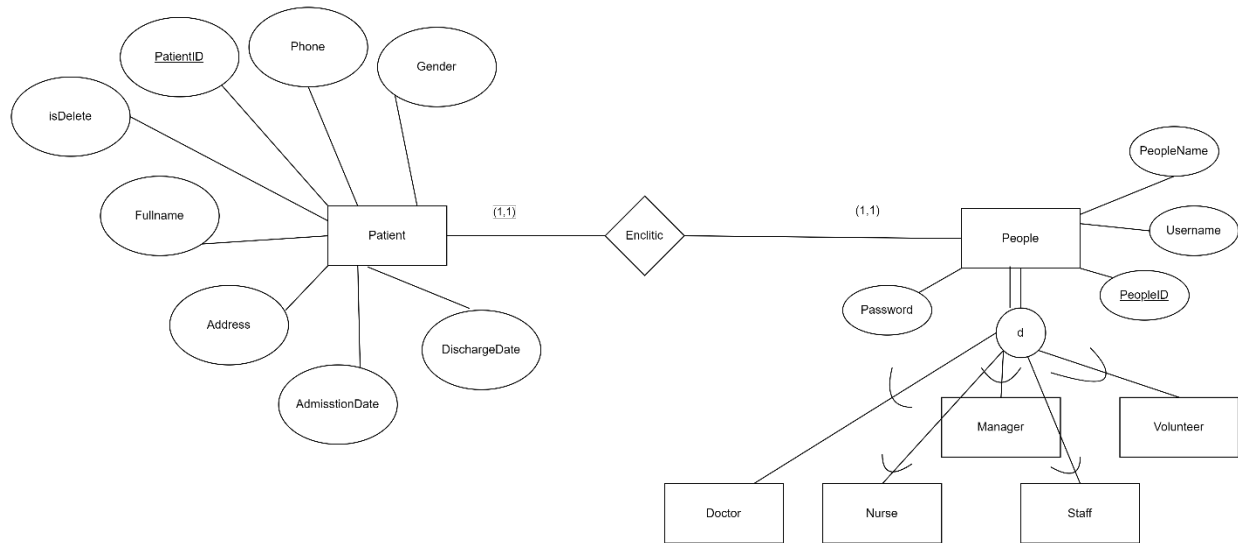
- So, it is important to track a patient location history. The camp needs to know the admission date, from where the patient is moved to the camp, the staff information, and the testing information if any.



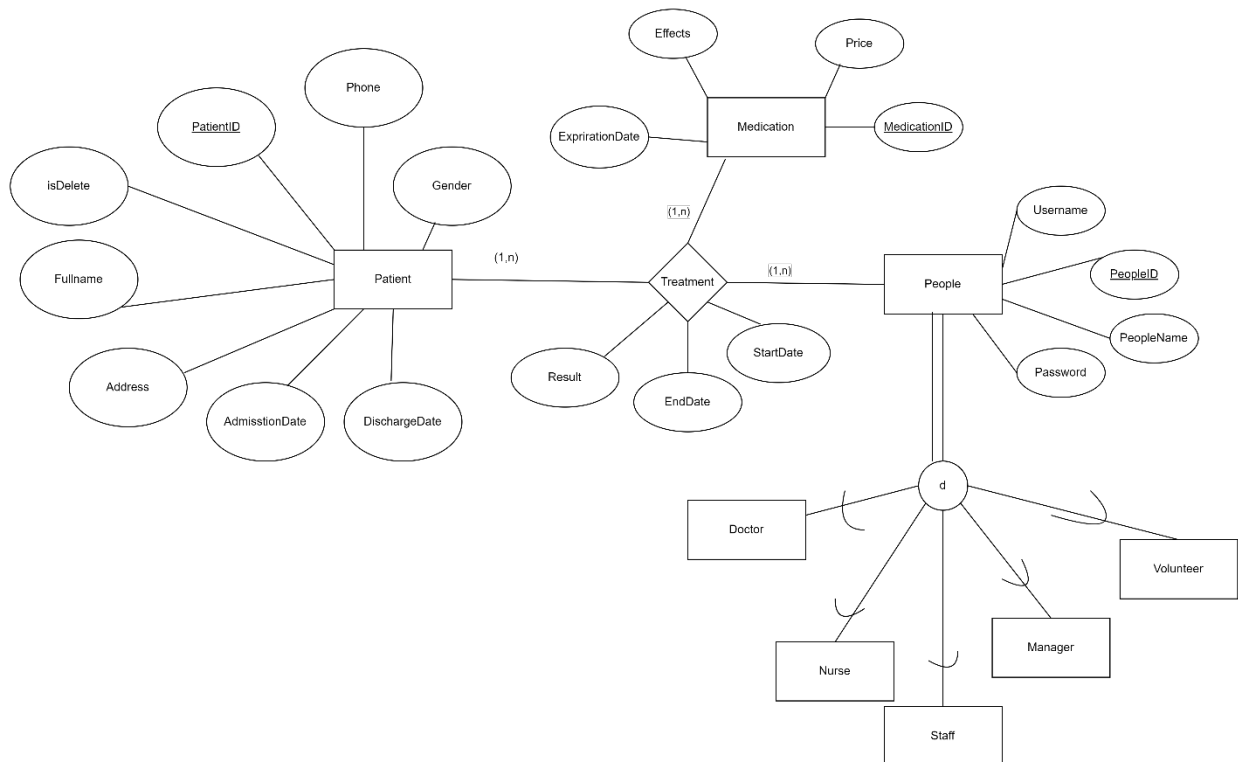
- Besides, the camp has several buildings, each has many floors and rooms. Each room has a limited capacity. There are three types of room: normal room, emergency room, and recuperation room. When admitted by a staff, a patient is assigned into a room based on his or her current condition.



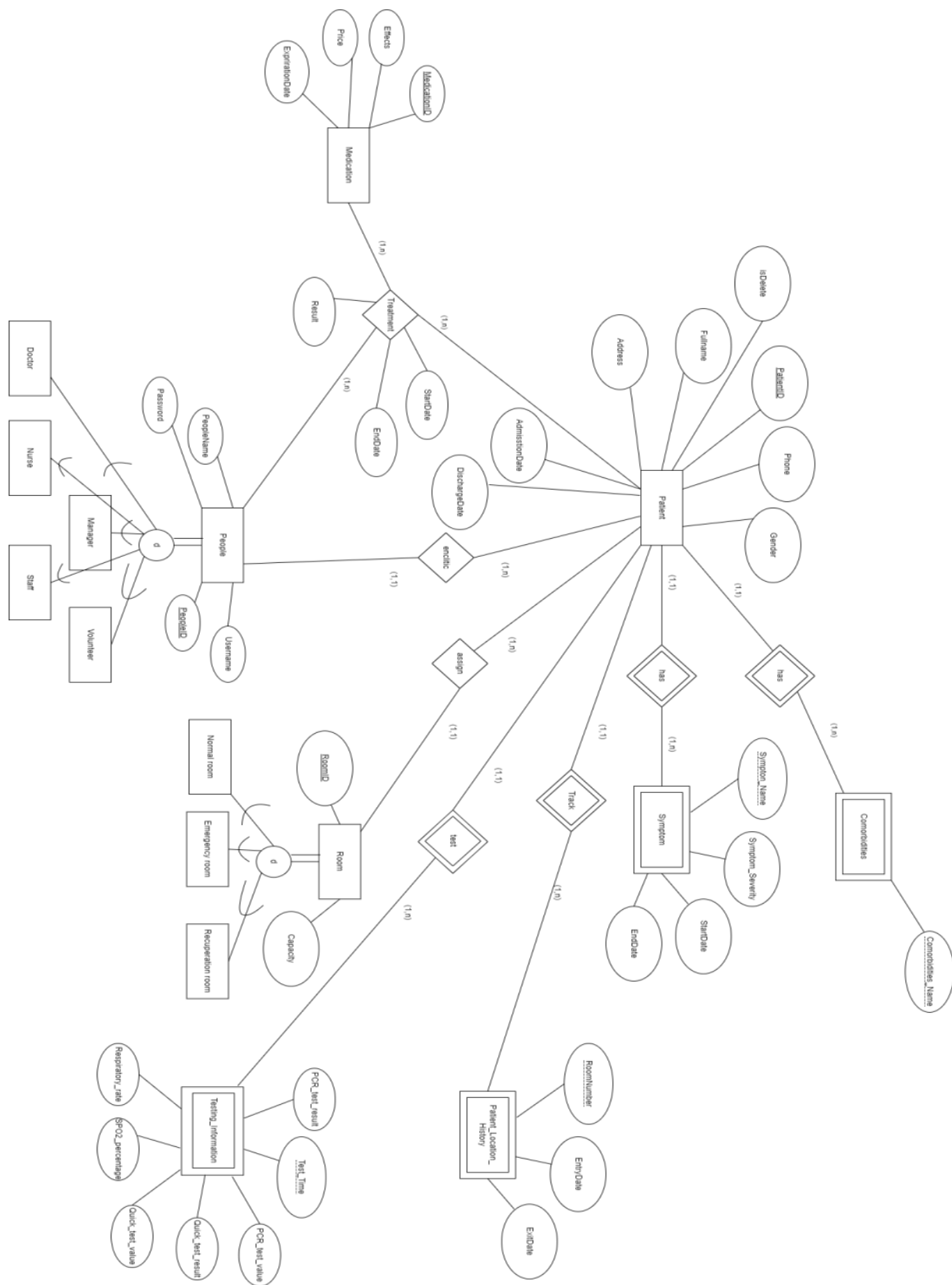
-A staff may admit many patients, and a patient is admitted by a staff. A patient can receive treatment from at least one doctor. A doctor can treat many patients at the same time, or sometimes, he has no patients to treat.



- The camp needs the details of each treatment such as: treatment period (start date and end date), result, and medications. Each patient is taken care of by a nurse; a nurse can take care of many inpatients at the same time. The information of a medication is also stored in the database. This information consists of a unique code, name of the medication, effects, price, and expiration date.



3.2.3 Mô hình tổng quát dữ liệu thực thể liên kết



CHƯƠNG IV: CHUYỂN TỪ MÔ HÌNH THỰC THỂ LIÊN KẾT SANG MÔ HÌNH QUAN HỆ

4.1 Cách chuyển đổi

- Bước 1: Ánh xạ các kiểu thực thể thông thường

Chuyển đổi các tập thực thể: Mỗi thực thể được chuyển thành một quan hệ cùng tên và danh sách thuộc tính. Đối với kiểu thực thể thông thường (regular entity type): Khóa của quan hệ là khóa của kiểu thực thể. Thuộc tính của quan hệ cũng chính là thuộc tính của kiểu thực thể. Quan hệ không chứa các thuộc tính đa trị mà chỉ chứa thuộc tính thành phần của những thuộc tính phức hợp.

- Bước 2: Ánh xạ các thực thể yếu

Đối với mỗi thực thể yếu W trong lược đồ ER cùng với kiểu thực thể chủ E , tạo một quan hệ R và bao gồm tất cả các thuộc tính đơn (hoặc thành phần đơn của thuộc tính phức hợp) của W là thuộc tính của R . Đưa các thuộc tính khóa chính của các quan hệ tương ứng với kiểu dữ liệu thực thể chủ làm khóa ngoại của R . Các thuộc tính này sẽ xác định kiểu liên kết của W . Khóa chính của R là một tổ hợp của khóa chính của các quan hệ tương ứng với kiểu thực thể chủ và khóa bộ phận của kiểu thực thể yếu W nếu có.

- Bước 3: Ánh xạ các loại mối quan hệ nhị phân 1:1

Với mỗi kiểu liên kết 1:1 R trong lược đồ ER, hãy xác định các quan hệ S và T tương ứng với các kiểu thực thể tham gia trong R . Hãy chọn một trong các quan hệ, chẳng hạn S , đưa khóa chính của T vào làm khóa ngoại trong S . Tốt nhất là chọn S là một kiểu thực thể tham gia toàn bộ vào R . Đưa tất cả các thuộc tính đơn (hoặc thành phần đơn của thuộc tính phức hợp) của kiểu liên kết 1:1 vào làm các thuộc tính của S .

Chú ý : Có một cách chuyển đổi mối liên kết 1:1 nữa là nhập hai kiểu thực thể và mỗi liên kết thành một quan hệ cách này thường được áp dụng khi cả hai kiểu vật thể đều tham gia toàn bộ vào liên kết

- Bước 4: Ánh xạ các loại mối quan hệ nhị phân 1:N

Với mỗi kiểu liên kết hai ngôi R kiểu 1:N, hãy xác định quan hệ S biểu diễn kiểu thực thể tham gia ở phía N của kiểu liên kết. Đưa khóa chính của quan hệ T biểu diễn kiểu thực thể tham gia vào R ở phía 1 vào làm khóa ngoại trong S làm như vậy vì mỗi thực thể cụ thể của phía N được liên kết với nhiều nhất là một thực thể cụ thể của phía 1 của kiểu liên kết. Đưa các thuộc tính đơn (hoặc các thành phần đơn của các thuộc tính phức hợp) của kiểu liên kết

1:N vào các thuộc tính của S

- Bước 5: Ánh xạ các kiểu quan hệ M:N nhị phân

Với mỗi kiểu liên kết N:M 2 ngôi R hãy tạo ra một quan hệ mới S để biểu diễn R. Đưa các khóa chính của các quan hệ biểu diễn các kiểu thực thể tham gia vào khóa ngoại của S. Tổ hợp các khóa chính đó sẽ tạo nên một khóa chính của S. Đưa tất cả các thuộc tính đơn (hoặc các thành phần đơn của thuộc tính phức hợp) của kiểu liên kết N:M vào làm thuộc tính của S chú ý ta không thể biểu diễn một kiểu liên kết N:M bằng một thuộc tính khóa ngoại đơn giản trong một trong các quan hệ tham gia như đã làm với các kiểu liên kết 1:1 và 1:N.

-Bước 6: Ánh xạ các thuộc tính đa trị

Với mỗi thuộc tính đa giá trị A, tạo một thuộc tính mới quan hệ R. Quan hệ R này sẽ bao gồm một thuộc tính tương ứng với A, cộng với thuộc tính khóa chính K-là khóa ngoại trong R-của mỗi quan hệ đại diện cho loại thực thể hoặc loại mối quan hệ có A là thuộc tính. Khóa chính của R là sự kết hợp của A và K. Nếu thuộc tính đa giá trị là phức hợp, chúng ta bao gồm các thành phần đơn giản của nó

- Bước 7: Ánh xạ các kiểu quan hệ N-ary

Đối với mỗi mối quan hệ n-ary loại R, trong đó $n > 2$, tạo một mối quan hệ mới S để đại diện cho R. Bao gồm các thuộc tính khóa ngoại trong S là các khóa chính của các mối quan hệ đại diện cho thực thể tham gia các loại. Cũng bao gồm bất kỳ thuộc tính đơn giản nào của n-ary loại mối quan hệ (hoặc các thành phần đơn giản của hỗn hợp thuộc tính) là thuộc tính của S.

4.2 Xác định các bảng và thuộc tính của bảng:

+Patient(PatientID,RoomID,PeopleID,Fullname,Address,Phone,Gender,AdmisstionDate,DischargeDate,isDelete)

Patient
<u>PatientID</u>
<u>RoomID</u>
<u>PeopleID</u>
FullName
Address
Phone
Gender
AdmissionDate
DischargeDate
isDelete

- Comorbidities(PatientID,Comorbidities_Name)

Comorbidities
<u>PatientID</u>
<u>Comorbidities_Name</u>

- Sympton(PatientID,Sympton_Name,Sympton_Severity,StartDate,EndDate)

Sympton
<u>PatientID</u>
<u>Sympton_Name</u>
Sympton_Severity
StartDate
EndDate

- Patient_Location_History(PatientID,RoomNumber,EntryDate,ExitDate)

Patient_Location_History
<u>PatientID</u> <u>RoomNumber</u> EntryDate ExitDate

+Testing_information(PatientID,Test_Time,PCR_test_result,PCR_test_value,Quick_test_result,Quick_test_value,SPO2_percentage,Respiratory_rate)

Testing_information
<u>PatientID</u> <u>Test_Time</u> PCR_test_result PCR_test_value Quick_test_result Quick_test_value SPO2_percentage Respiratory_rate

- Room(RoomID,Capacity)

Room
<u>RoomID</u> Capacity

- People(PeopleID,PeopleName,Username,Password)

People
<u>PeopleID</u> PeopleName Username Password

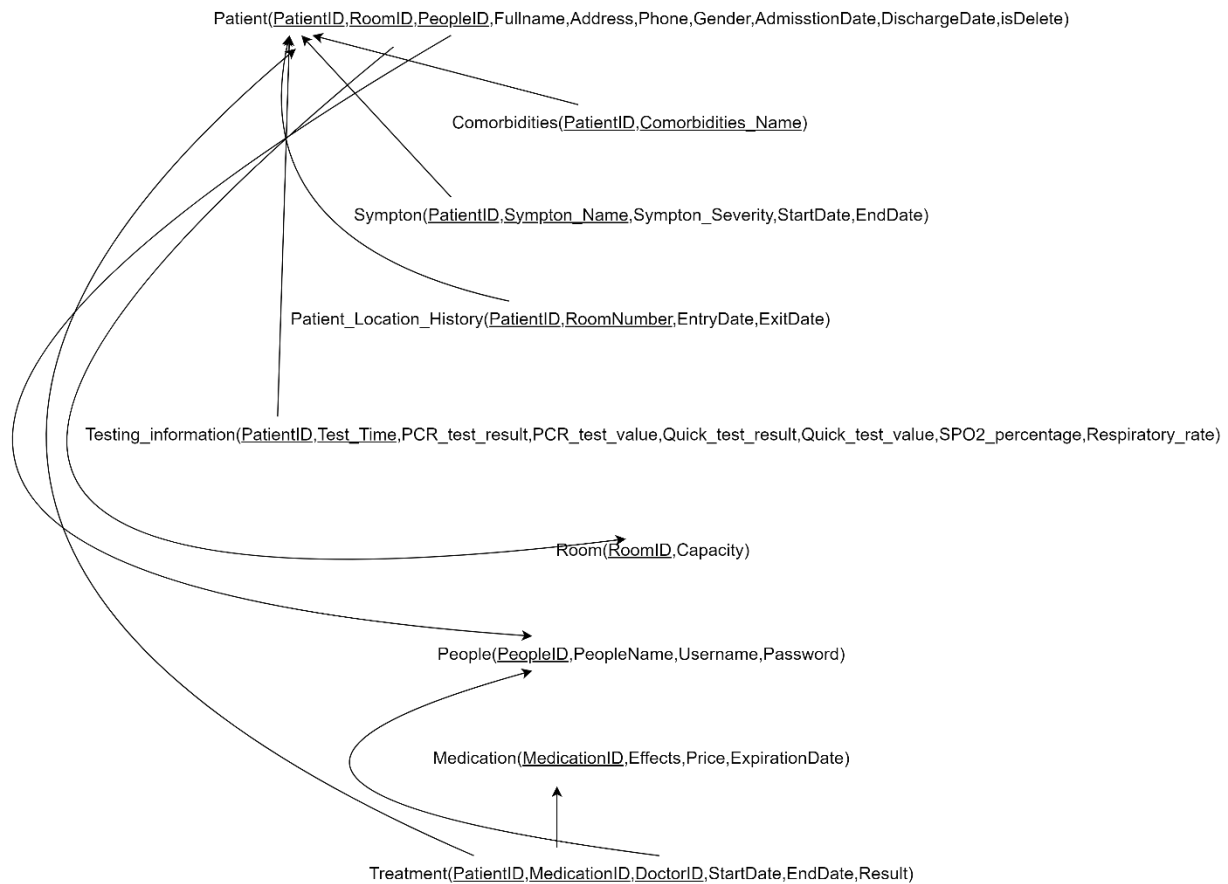
- Medication(MedicationID,Effects,Price,ExpirationDate)

Medication
<u>MedicationID</u>
Effects
Price
ExpirationDate

- Treatment(PatientID,MedicationID,DoctorID,StartDate,EndDate,Result)

Treatment
<u>PatientID</u>
<u>MedicationID</u>
<u>DoctorID</u>
StarDate
EndDate
Result

4.3 Mô hình dữ liệu quan hệ



CHƯƠNG V: ÁP DỤNG CSDL TRÊN MÔI TRƯỜNG WEBSITE

5.1 Các ngôn ngữ sử dụng

- **PHP**: Được sử dụng để xử lý các yêu cầu máy chủ và tương tác với cơ sở dữ liệu. Giúp tạo ra nội dung động và thực hiện các chức năng như: đăng nhập, đăng ký, liệt kê thông tin,...
- **HTML**: Được sử dụng để xây dựng cấu trúc siêu văn bản trên một website, hay khai báo các tập tin kỹ thuật số như: hình ảnh, đoạn văn bản, hình ảnh, liên kết, biểu mẫu,...
- **JavaScript**: Được sử dụng để thêm các tính năng động và tương tác người dùng vào trang web. Nó có thể thực hiện các hành động như kiểm tra dữ liệu đầu vào, thay đổi nội dung trang mà không cần tải lại, và thực hiện các hiệu ứng hình ảnh.
- **CSS**: Được sử dụng để kiểm soát giao diện và trình bày của trang web, bao gồm màu sắc, font chữ, kích thước, và bố cục.
- **SQL**: Được sử dụng để truy vấn, cập nhật, và quản lý dữ liệu trong cơ sở dữ liệu.

5.2. Các Framework sử dụng

- **Bootstrap**: Giúp phát triển giao diện người dùng web. Bao gồm HTML, CSS, và JavaScript để giúp xây dựng trang web thân thiện với di động và hiện đại.

5.3. Sơ lược về website

5.3.1 Kịch bản:

Do dịch Covid-19 bùng phát, một trại cách ly đã được thiết lập để cách ly và theo dõi những người bị điều tra, website này dùng để quản lý bệnh nhân, quản lý các bác sĩ cũng như những người làm việc trong trại cách ly này, quản lý thuốc và các phòng cho bệnh nhân

5.3.2 Cách cài đặt:

Có thể cài đặt thông qua github.

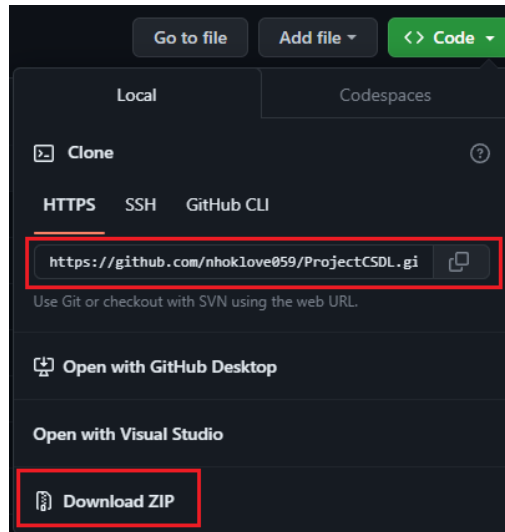
B1: Truy cập github.com/nhoklove059/ProjectCSDL

B2: Nhấn vào nút thì sẽ có



một cửa sổ hiện ra

B3: Có thể tải về thông qua 2 cách là Download ZIP hoặc sử dụng git clone với URL trên

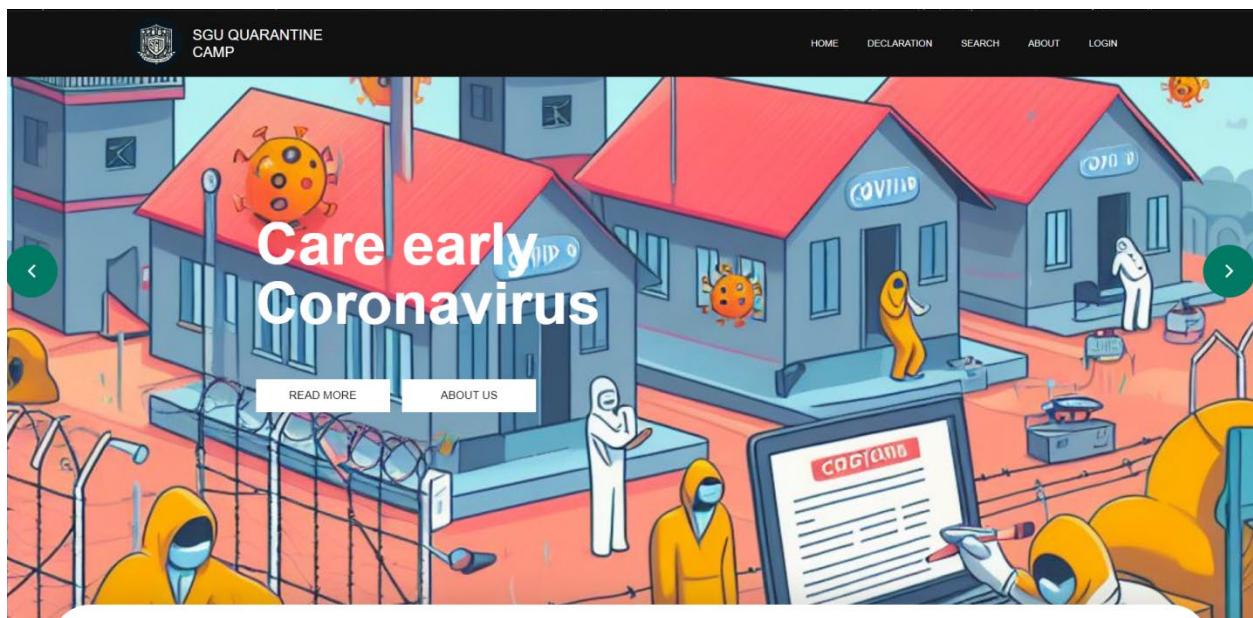


B4: Bỏ vào tệp chứa ứng dụng cung cấp môi trường WAMP mà các bạn đang sử dụng, như mình đang sử dụng Laragon thì sẽ bỏ tệp project đã tải về theo đường dẫn laragon\www

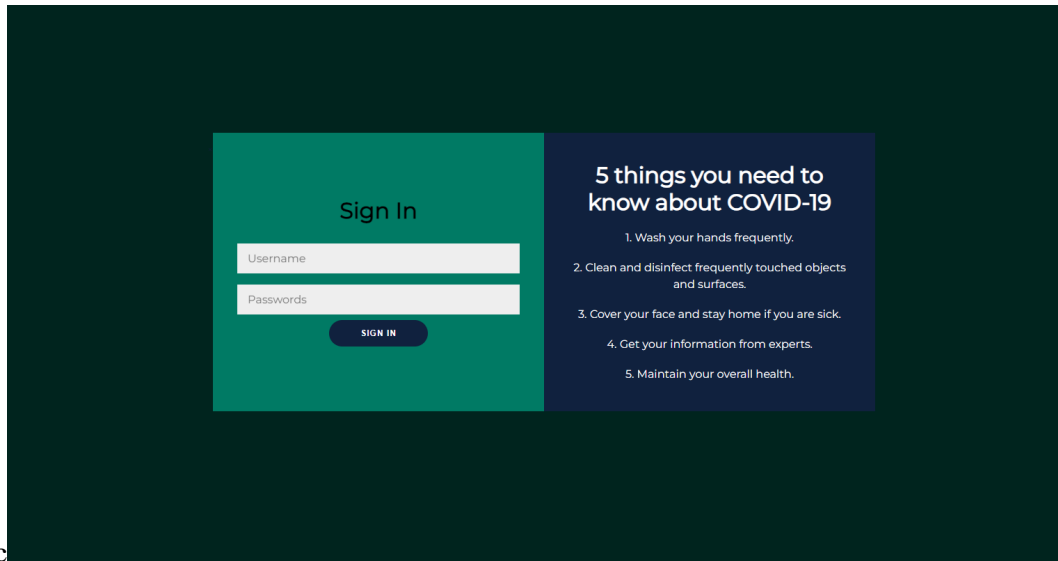
B5: Start Apache để bắt đầu truy cập vào trang web

5.3.3 Sơ lược về Website

- Khi bắt đầu truy cập vào website thì bạn sẽ được đưa vào trang chào mừng và sơ lược thông tin

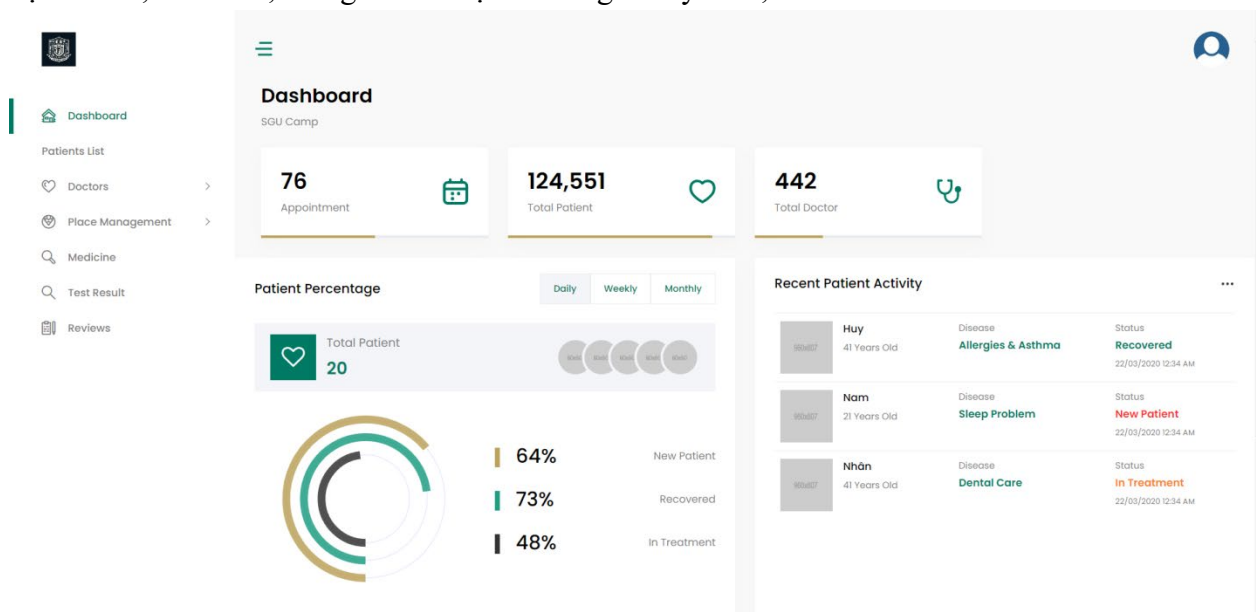


- Để bắt đầu nghiệp vụ quản lý bệnh nhân thì bạn sẽ đăng nhập bằng tài khoản đã được cung

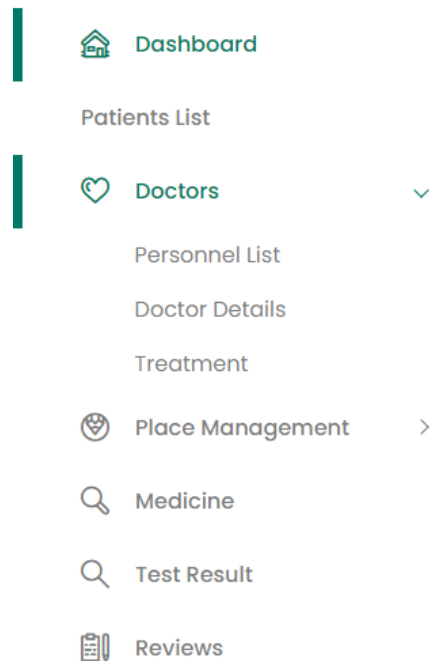


cấp trước

- Sau khi đăng nhập thì bạn sẽ được đưa tới trang chính với các thông tin sơ lược về trại như số bệnh nhân, số bác sĩ, thông tin các bệnh nhân gần đây nhất,



- Sử dụng thanh menu bên trái để đưa tới các trang thông tin mà bạn muốn xem



1. Trang Patient List hiển thị thông tin sơ lược về các bệnh nhân như ID, tên, bác sĩ chăm sóc, liên hệ, ...

Patient
Patients List

+New Patient

Profile Datable

Show 10 entries Search:

ID	Name	Gender	Doctor Assigned	Contact	Comorbidities	Symptoms	Room	Action
1	John Doe	Male	Doctor 1	555-1234	Diabetes	Fever Cough	R1	
2	Jane Smith	Female	Doctor 1	555-5678	Diabetes Asthma	Fever	R2	
3	Bob Johnson	Male	Doctor 1	555-9876	Chronic Lung Disease Heart Condition	Cough	R3	
4	Alice Brown	Female	Doctor 1	555-4321	Heart Condition Chronic Lung Disease	Fever	R1	
5	David Davis	Male	Doctor 1	555-2468	Immunocompromised State Diabetes	Cough	R2	
6	Eva White	Female	Doctor 2	555-9753	Chronic Lung Disease Asthma	Fever	R3	

- Tìm kiếm các bệnh nhân ở mục “Search” theo họ tên.

Profile Datatable

Show

10

entries

Search:

John

ID	Name	Gender	Doctor Assigned	Contact	Comorbidities	Symptoms	Room	Action
1	John Doe	Male	Doctor 1	555-1234	Diabetes	Fever Cough	R1	<div></div> <div></div> <div></div>
3	Bob Johnson	Male	Doctor 1	555-9876	Chronic Lung Disease Heart Condition	Cough	R3	<div></div> <div></div> <div></div>

Showing 1 to 2 of 2 entries (filtered from 21 total entries)

Previous

1

Next

2. Bạn có thể thêm thông tin bệnh nhân mới bằng nút “+ New Patient”

Add Patient

Patient Name

Full name

Gender

Choose Gender ...

Doctor

None

Comorbidities

To select multiple, hold down the Ctrl key and select

Older Age

Lung problems, including asthma

Heart disease

Brain and nervous system conditions

Diabetes and obesity

Symptoms

To select multiple, hold down the Ctrl key and select

Cough

Headache

Fatigue


Shortness of breath or difficulty breathing


Muscle or body aches


Room



Khi nhấn vào một bệnh nhân bất kì thì trang web sẽ chuyển hướng tới trang thông tin chi tiết của bệnh nhân và các nút chức năng như chỉnh sửa thông tin bệnh nhân, xoá, in báo cáo,...

My experiences, along with the combined testimonies of trans and rare disease patients around the world, prove a throughline within medicine that I believe is getting in the way of a healthier experience for everyone: that medical professionals are too often on auto-pilot, looking for horses when faced with zebras, and not willing enough to learn from their patients' self-reported experiences. I would go as far as to add that those with the specialized knowledge necessary to diagnose rare diseases are seemingly even less likely to have an up-to-date education on bedside manner that is inclusive of trans and gender-nonconforming people.


 Address
123 Main St
CADGE 94107

 Phone
555-1234

 Email
info@mail.com



John Doe
Symptom: Fever, Cough
Comorbidity: Diabetes



Story About Disease

3. Bên dưới có các thông tin về xét nghiệm, chăm sóc và lịch sử di chuyển của bệnh nhân.

Testing									
Show 10 entries		Search: <input type="text"/>							
Test Number	Patient	Test Date	Status	PCR Test	PCR Value	Quick Test	Quick Test Value	SP02	Respiratory rate
1	John Doe	2023-09-01	Warning	Positive	25	Negative	None	94%	22 bpm
2	John Doe	2023-09-01	Wait for test	Negative	None	Positive	23	10%	11 bpm
3	John Doe	2023-09-01	Wait for test	Negative	None	Positive	23	10%	11 bpm
Showing 1 to 1 of 1 entries							<div>Previous1Next</div>		


Treatment			
Show	10	entries	Search: <input type="text"/>
Doctor	Patient	Date Start	Date End
Doctor 1	John Doe	2023-09-01 00:00:00	Unknown
Showing 1 to 1 of 1 entries			<div>Previous</div> <div>1</div> <div>Next</div>

Location History			
Show	10	entries	Search: <input type="text"/>
Patient	Date Start	Date End	Room
John Doe	2023-09-01 10:00:00	Unknown	R1
Showing 1 to 1 of 1 entries			<div>Previous</div> <div>1</div> <div>Next</div>

4. Khi nhấn nút “Print” sẽ in ra thông tin của bệnh nhân, thông tin về các ca xét nghiệm và các ca điều trị.

SQL Quarantine Camp

03-12-2023



From:
SGU Quarantine Camp
273 An D. Vương
Phường 3, Quận 5, TP.HCM
Email: SGU@gmail.com
Phone: +48 444 666 3333

Patient Details

ID	Name	Gender	Doctor Assigned	Contact	Comorbidities	Symptoms	Room
1	John Doe	Male	Doctor 1	555-1234	Diabetes	Fever Cough	R1

Testing Results

Test ID	Patient	Test Date	Status	PCR Test	Quick Test	SP02	Respiratory rate
1	John Doe	2023-09-01	Warning	Positive	Negative	94%	22 bpm

Treatment

Doctor	Patient	Result	Date Start	Date End
Doctor 1	John Doe	Recovered	2023-09-02	2023-09-05

- Mục Doctors sẽ bao gồm các trang như xem thông tin những người làm việc trong trại cách ly, chi tiết thông tin bác sĩ (Chỉ có tài khoản cung cấp cho bác sĩ mới được xem) và lịch sử chăm sóc các bệnh nhân của bác sĩ tương ứng

ID	Name	Job type
D1	Doctor 1	Doctor
D2	Doctor 2	Doctor
M1	Manager	Manager
N1	Nurse 1	Nurse
N2	Nurse 2	Nurse
S1	Staff 1	Staff
S2	Staff 2	Staff
V1	Volunteer 1	Volunteer
V2	Volunteer 2	Volunteer
V3	Volunteer 3	Volunteer

Doctor 1
#D1
Join on 21 August 2020, 12:45 AM

Biography
Every cancer patient, as well as their family, is different and has specific requirements. I aim to create a customized treatment plan for each and every one of my patients to help them achieve the best results.

Appointment
December 2023

Su	Mo	Tu	We	Th	Fr	Sa
26	27	28	29	30	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31	1	2	3	4	5	6

Wednesday, June 3th
08:00 - 10:30 AM
Dr. Samantha

Tuesday, June 16th
08:00 - 10:30 AM

ID	Name	Join Date	Status
D1	Bob Johnson	2023-09-06	Recovered from moderate cough
D1	Alice Brown	2023-09-02	Recovered from fever

- Mục quản lý các phòng và tình trạng của phòng bệnh. Ở đây quản lý có thể thêm các phòng mới bằng nút “+Room”

- Dashboard
- Patients List
- Doctors
- Place Management
 - Room
 - Medicine
 - Test Result
 - Reviews

Room

Room Management

+Room
+Floor
+building

	Room ID	Room Type	Capacity	Status	
<input type="checkbox"/>	R1	Normal	10	FULL	⋮
<input type="checkbox"/>	R10	Normal	10	FULL	⋮
<input type="checkbox"/>	R2	Emergency	5	FREE	⋮
<input type="checkbox"/>	R3	Recuperation	8	ALMOST FULL	⋮
<input type="checkbox"/>	R4	Normal	10	FULL	⋮
<input type="checkbox"/>	R5	Emergency	5	FREE	⋮
<input type="checkbox"/>	R6	Recuperation	8	ALMOST FULL	⋮

- Mục quản lý thuốc, giá thành và thời hạn. Ở đây quản lý có thể thêm các thuốc mới bằng nút “+ New Medicine”

- Dashboard
- Patients List
- Doctors
- Place Management
 - Room
 - Medicine
 - Test Result
 - Reviews

Medicine

Medicines Management

+New Medicine
Filter Date

	Medication ID	Medication Detail	Effects	Price	Expiration Date	
<input type="checkbox"/>	Med1	Paracetamol	Pain relief	10.50 vnd	2023-12-01	⋮
<input type="checkbox"/>	Med10	Atorvastatin	Cholesterol medication	13.75 vnd	2024-01-05	⋮
<input type="checkbox"/>	Med2	Aspirin	Fever reducer	8.75 vnd	2024-01-15	⋮
<input type="checkbox"/>	Med3	Ciprofloxacin	Antibiotic	15.20 vnd	2023-11-20	⋮
<input type="checkbox"/>	Med4	Ibuprofen	Anti-inflammatory	12.30 vnd	2023-12-10	⋮
<input type="checkbox"/>	Med5	Omeprazole	Heartburn relief	9.90 vnd	2023-12-05	⋮
<input type="checkbox"/>	Med6	Amoxicillin	Antibiotic	18.40 vnd	2023-11-25	⋮
<input type="checkbox"/>	Med7	Loratadine	Antihistamine	7.60 vnd	2024-02-01	⋮
<input type="checkbox"/>	Med8	Hydrochlorothiazide	Diuretic	14.80 vnd	2023-11-15	⋮
<input type="checkbox"/>	Med9	Metformin	Diabetes medication	11.25 vnd	2024-03-01	⋮

5. Index và các trường hợp sử dụng index:

- Trong trường hợp có 15 triệu dòng dữ liệu trong bảng “Patient”. Nhóm em đã sử dụng câu lệnh SQL:

```
SELECT * FROM patient WHERE fullName = “Name52”
```

- Cách tạo index:

```
CREATE INDEX index_name ON table_name(column_name)
```

- Trong trường hợp của nhóm em:

```
CREATE INDEX IDX_FULLNAME ON patient(fullName)
```

- Sau đây là kết quả sau khi sử dụng index để tìm kiếm thay vì tìm kiếm thông thường:

- Index:

id	select_type	table	type	possible_keys	key	key_len	ref	rows	Extra
1	SIMPLE	patient	ref	IDX_FULLNAME	IDX_FULLNAME	138	const	305610	Using index condition

✓ Showing rows 0 - 24 (159273 total, Query took 0.0004 seconds.)

```
SELECT * FROM patient WHERE fullName = "Name52";
```

- Không index:

id	select_type	table	type	possible_keys	key	key_len	ref	rows	Extra
1	SIMPLE	patient	ALL	NULL	NULL	NULL	NULL	15483374	Using where

✓ Showing rows 0 - 24 (159273 total, Query took 0.0036 seconds.)

```
SELECT * FROM patient WHERE fullName = "Name52";
```

- Chứng minh:

- $r = 15000000$, $R = 100$ bytes, $B = 4096$ bytes
- $bfr = 4096/40 = 40$
- $b = 15000000/40 = 375000$
- A non-ordering key field of the file that is $V = 9$ bytes long
- A linear search on the file would require $b/2 = 375000/2 = 187500$ blocks accesses on the average.
- A secondary index
 - A block pointer is $P = 6$ bytes long, so each index entry is $R_i = (9 + 6) = 15$ bytes.
 - $bfr_i = \lfloor (B/R_i) \rfloor = \lfloor (4,096/15) \rfloor = 273$ index entries per block
 - $b_i = \lceil (r/bfr_i) \rceil = \lceil (15000000/273) \rceil = 54946$ blocks
 - A binary search on this secondary index needs $\lceil (\log_2 b_i) \rceil = \lceil (\log_2 54946) \rceil = 16$ block accesses
 - Total block accesses: $16 + 1 = 17$ blocks

6. Bảo mật và cách phòng chống tấn công cơ sở dữ liệu

- Bảo mật cơ sở dữ liệu (CSDL) là quá trình đảm bảo rằng dữ liệu được bảo vệ khỏi truy cập

trái phép hoặc hư hỏng, bao gồm các cuộc tấn công từ bên ngoài và bên trong, cũng như các mối đe dọa từ bên trong tổ chức. Điều này đòi hỏi việc sử dụng các công nghệ, chính sách và quy trình để bảo vệ CSDL khỏi các cuộc tấn công phá hoại, các hành vi sử dụng sai hoặc sai sót của con người.

- Việc bảo mật CSDL rất quan trọng vì nhiều lý do:
 - + Đầu tiên, nó ngăn chặn truy cập trái phép vào dữ liệu.
 - + Thứ hai, nó giúp giảm thiểu các lỗi do người dùng.
 - + Thứ ba, nó đảm bảo rằng thông tin không bị mất hoặc bị thay đổi một cách không mong muốn.
 - + Thứ tư, nó giữ cho nội dung dữ liệu và chương trình xử lý không bị tiết lộ. Cuối cùng, với sự phát triển của kết nối mạng, vai trò của bảo mật CSDL ngày càng tăng.
- Một số phương pháp tấn công CSDL:
 - + **SQL Injection:** Đây là một kiểu tấn công mạng phổ biến, trong đó kẻ tấn công chèn các mã SQL query/command vào ô input dữ liệu trước khi chuyển cho ứng dụng web xử lý
 - Ví dụ: Khi sử dụng truy vấn để kiểm tra login

```
statement = "SELECT * FROM users WHERE name = '" + userName + "';"
```

Tin tặc có thể nhập a' or 't'='t hoặc là a' -- ở giá trị biến username thì câu lệnh sẽ biến thành như sau

```
SELECT * FROM users WHERE name = 'a' or 't'='t';
```

```
SELECT * FROM users WHERE name = 'a' -- ;
```

- Một số cách phòng chống SQL Injection:
 - + Không tin tưởng kênh Input của người dùng
 - + Xác nhận chuỗi các input ở phía máy chủ.
 - + Sử dụng các câu lệnh tham số.
 - + Ngăn chặn SQLi on MYSQL: Sử dụng chuỗi mysql_real_escape cho các truy vấn

+**DDos:** Đây là một kiểu tấn công mạng bằng cách gây “ngập lụt” đối tượng bằng các gói UDP và ICMP

- Một số cách phòng chống DDos:
 - + Tăng băng thông
 - + Chuyển hướng các cuộc tấn công.
 - + Sử dụng cơ chế chống mạo danh IP
 - + Giới hạn số lượng truy cập

- Một số cách phòng tránh tấn công CSDL:

peopleName	role	username	password
Doctor 1	Doctor	doc1	827ccb0eea8a706c4c34a16891f84e7b
Doctor 2	Doctor	doc2	827ccb0eea8a706c4c34a16891f84e7b
HUY	Admin	admin	827ccb0eea8a706c4c34a16891f84e7b
Nurse 1	Nurse	nur1	827ccb0eea8a706c4c34a16891f84e7b
Nurse 2	Nurse	nur2	827ccb0eea8a706c4c34a16891f84e7b
HULL	HULL	HULL	
Staff 1	Staff	sta1	827ccb0eea8a706c4c34a16891f84e7b
Staff 2	Staff	sta2	827ccb0eea8a706c4c34a16891f84e7b
Volunteer 1	Volunteer	vol1	827ccb0eea8a706c4c34a16891f84e7b
Volunteer 2	Volunteer	vol2	827ccb0eea8a706c4c34a16891f84e7b
Volunteer 3	Volunteer	vol3	827ccb0eea8a706c4c34a16891f84e7b

+ Phân

quyền truy

cập và nhận dạng người dùng: Người dùng được phân quyền truy cập CSDL dựa trên vai trò của họ. Bảng phân quyền truy cập là một phần của CSDL và được quản lý chặt chẽ. Hệ thống CSDL cần phải nhận biết được người dùng, giải pháp là dùng mật khẩu hoặc chữ ký điện tử.

+ **Mã hóa thông tin và nén dữ liệu:** Các thông tin quan trọng và nhạy cảm thường được lưu trữ dưới dạng mã hóa để giảm khả năng rò rỉ. Có nhiều cách mã hóa khác nhau, tiêu biểu là nén dữ liệu để giảm dung lượng bộ nhớ lưu trữ dữ liệu.

+ **Lưu log:** Log hệ thống cho biết số lần truy cập vào hệ thống, vào từng thành phần của hệ thống, vào từng yêu cầu tra cứu, thông tin về số lần cập nhật cuối cùng. Log hệ thống hỗ trợ cho việc khôi phục hệ thống khi có sự cố kỹ thuật, cung cấp thông tin cho phép đánh giá mức độ quan tâm của người dùng đối với hệ thống.

CHƯƠNG VI: ĐÁNH GIÁ VÀ KẾT LUẬN

Đánh Giá Tiến Độ Hoàn Thành Đồ Án:

1. Hoàn Thành Các Câu Hỏi (1, 2, 3):

- Nhóm đã hoàn thành đầy đủ các câu hỏi được giao trong yêu cầu của đề bài, bao gồm các phần : 3.1, 3.2, 3.3, 3.4, 3.5, 3.6.
- Các câu trả lời có vẻ đầy đủ và có sự chi tiết, đáp ứng được nhu cầu của đề bài.

2. Chất Lượng Công Việc:

- Các câu trả lời có tính logic và có sự kết hợp giữa lý thuyết và ví dụ cụ thể.
- Sử dụng ngôn ngữ chuyên nghiệp và chính xác trong việc diễn đạt ý.

3. Thời Gian Hoàn Thành:

- Việc hoàn thành theo tiến độ là một điểm tích cực, thể hiện sự tổ chức và quản lý thời gian hiệu quả từ phía nhóm.

4. Tương Tác Nhóm:

- Có sự tương tác tích cực và hiệu quả giữa các thành viên nhóm, điều này giúp đảm bảo rằng công việc được thực hiện một cách hợp nhất.

Kết Luận:

Dựa trên thông tin trên, có thể kết luận rằng nhóm đã thực hiện công việc một cách đầy đủ và

chất lượng. Việc hoàn thành theo tiến độ là một điểm mạnh, cho thấy sự cam kết và tổ chức tốt từ phía nhóm. Các câu trả lời đều mang lại sự hiểu biết sâu sắc về các khía cạnh của đề tài. Nhóm đã làm việc một cách đáng khen ngợi và đạt được mục tiêu đề ra trong đồ án.