A picture containing text, clipart

Description automatically generated

**Project Report**

|  |  |
| --- | --- |
| Module Name | WSQ Database Design and Implementation (SF) |
| Course Name | Postgraduate Diploma in Software Engineering |
| Assignment Title | Plan, Design, and Implement a Database for eCommerce Portal |

|  |  |  |  |
| --- | --- | --- | --- |
| Learner Name | | Assessor Name | |
| Huynh Minh Phu | | Mr. Giau, Mr. Lap, Mr. Tu | |
| Date Issued | Completion Date | | Submitted On |
| 15/12/2023 | 15/12/2023 | | 15/12/2023 |

**Plagiarism**

Plagiarism covers both copying or paraphrasing with the source without proper understanding and citation. The source includes but is not limited to:

* Books, Journal Articles, Online resources
* Others Assignment or project
* Software program
* Images

The learners must submit their work without plagiarism while producing the evidence against the assessment criteria. They must apply proper citation and Harvard referencing. Their work must go through the Turnitin software to check for the similarity index. Please consult the relevant module instructor or mentor if you need any further advice.

|  |
| --- |
| Learner declaration |
| I certify that the work submitted for this assignment is my own and research sources are fully acknowledged.  Learner signature: Phu Date: 15/12/2023 |

Table of Contents

[1. Project Background 3](#_Toc148867247)

[2. Project Objective 3](#_Toc148867248)

[3. Functional, Non-functional and Technical Requirements 3](#_Toc148867249)

[4. Task 1 3](#_Toc148867250)

[5. Task 2 3](#_Toc148867251)

[6. Task 3 3](#_Toc148867252)

[7. Task 4 4](#_Toc148867253)

[8. Task 5 4](#_Toc148867254)

[9. Task 6 4](#_Toc148867255)

## Project Background

The main objective of the project is to create a consumer-centric application with an optimized user experience. This requires an efficient database system capable of handling large amounts of data from three main user types: sellers, consumers, and administrators.

## Project Objective

This Assignment is used for the following:

• Summative Assessment of students in the Module ‘WSQ Database Design and

Implementation (SF)’ of the Course ‘Postgraduate Diploma in Software Engineering’.

In this assignment, you have to design, develop & implement a MySQL Database..

This assignment gives you an opportunity to demonstrate your capabilities in the

following areas:

• Apply critical thinking skills to analyse requirements and generate a database schema.

• Utilize analytical and problem-solving abilities to develop conceptual, logical, and

physical designs for implementing the database.

• Execute data population strategies to simulate business operations.

• Implement advance data manipulation techniques to generate management reports.

• Employ testing methodologies to ensure the accuracy, reliability, and optimal

performance of the database.

## Functional, Non-functional and Technical Requirements

**Project Scenario**

You currently work as a Data Engineer for Brightica design agency, where you design and

implement data models for client-centric products. As part of the role, your manager Mr.

Andrew assigned the project to develop an optimal database design to deliver Rich Internet

Application for Boutiqa. Boutiqa is a marketplace for sellers to promote their products and for

consumers to purchase with ease. The company wants to have a consumer-centric application

with an enhanced user experience.

**Project Overview**

You are required to demonstrate your capabilities in the following areas:

• Planning of database use group

• Conceptual, Logical, and Physical design of the database

• Writing queries and stored procedures to optimize the system performance and management reports.

The scope of the project in this module is to design and develop and implement the database.

The overview of the project is as below:

There are 3 types of users:

1. Sellers

2. Consumers

3. Administrator

Sellers should be able to perform following functions in the portal:

1. Register in the portal.

2. Update their Profile after logging in.

3. Maintain the product catalog to promote their products.

Consumers should be able to perform following functions in the portal:

1. Register in the portal.

2. Update their Profile after logging in.

3. Search products.

4. Choose products to view the details.

5. Add, edit, and remove items in the shopping cart.

Administrator should be able to perform following functions in the portal:

1. Administer user data.

2. Send bulk email invite to potential clients to register

## Task 1

* **Objective:** Outline the database's purpose for the Boutiqa project, focusing on a consumer-centric application with an enhanced user experience.
* **Scope:** The database must support functionalities for three user types: Sellers, Consumers, and Administrators.

**User Requirements**

* **Sellers:** Register, update profiles, manage product listings.
* **Consumers:** Register, update profiles, search and view products, manage shopping cart.
* **Administrators:** Manage user data, send bulk emails, and oversee the system.

**Data Requirements**

* **Entities:** Detail entities like User Accounts, Product Catalogs, Orders, etc.
* **Volume & Growth:** Estimate initial data volume and expected growth.

**Technical Specifications**

* **DBMS Choice:** Rationalize the choice of a specific DBMS (e.g., PostgreSQL, Oracle, etc.).
* **Security:** Outline encryption, data protection strategies.
* **Integration:** Specify necessary integrations (e.g., payment gateways, CRM systems).
* **Scalability:** Ensure the database can scale with user base growth.

**Compliance and Reporting**

**Regulations:** Address GDPR, CCPA, or other relevant data protection laws.

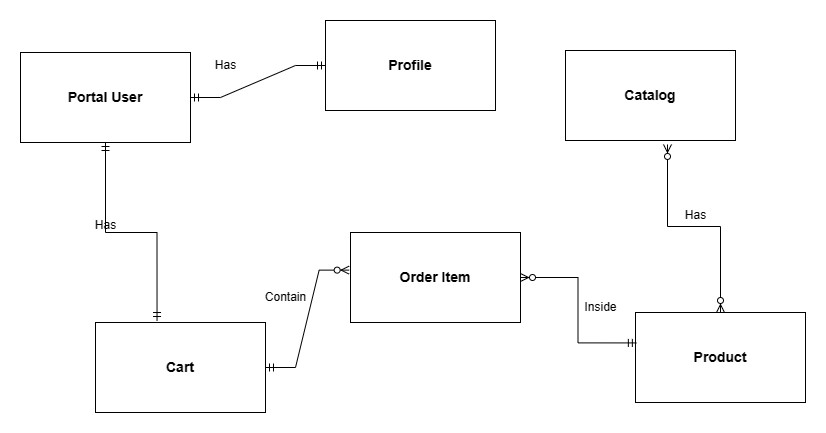
**Reporting Needs:** Define requirements for analytics and reporting capabilities.

## Task 2

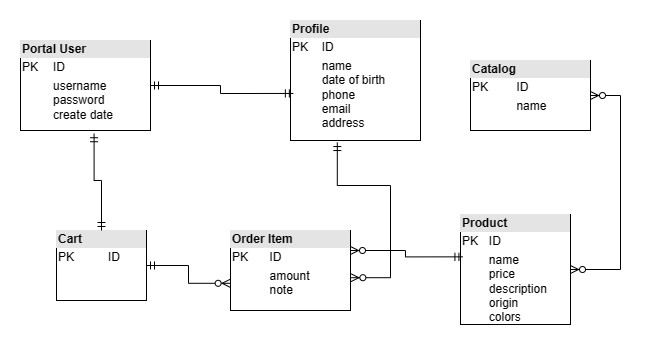
* List of entities with description

|  |  |
| --- | --- |
| **Entity Name** | **Attributes (what it contains)** |
| Product | * Name * Price * Description * origin * colors |
| Profile | * Name * Date of birth * Phone * Email * Address |
| Portal\_user | * Type: role of user (admin, seller, consumer) * Create date * Username * password |
| Catalog | * name |
| Order Item | * amount * note * order date |
| Cart | * cart info |

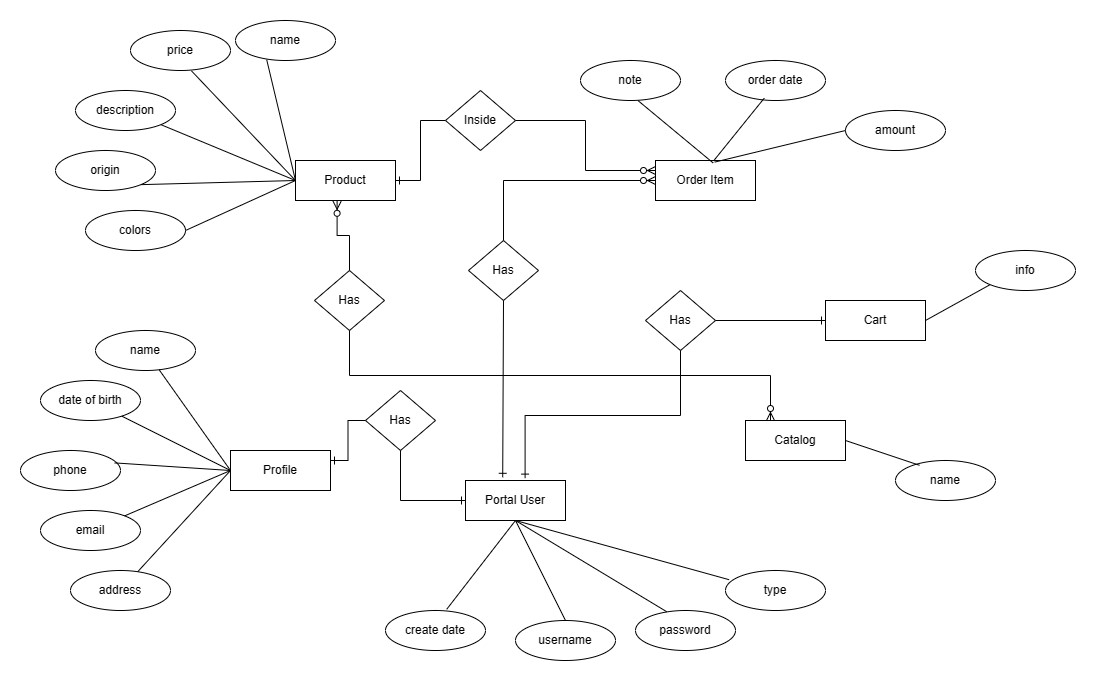
* Conceptual design



* Logical Design

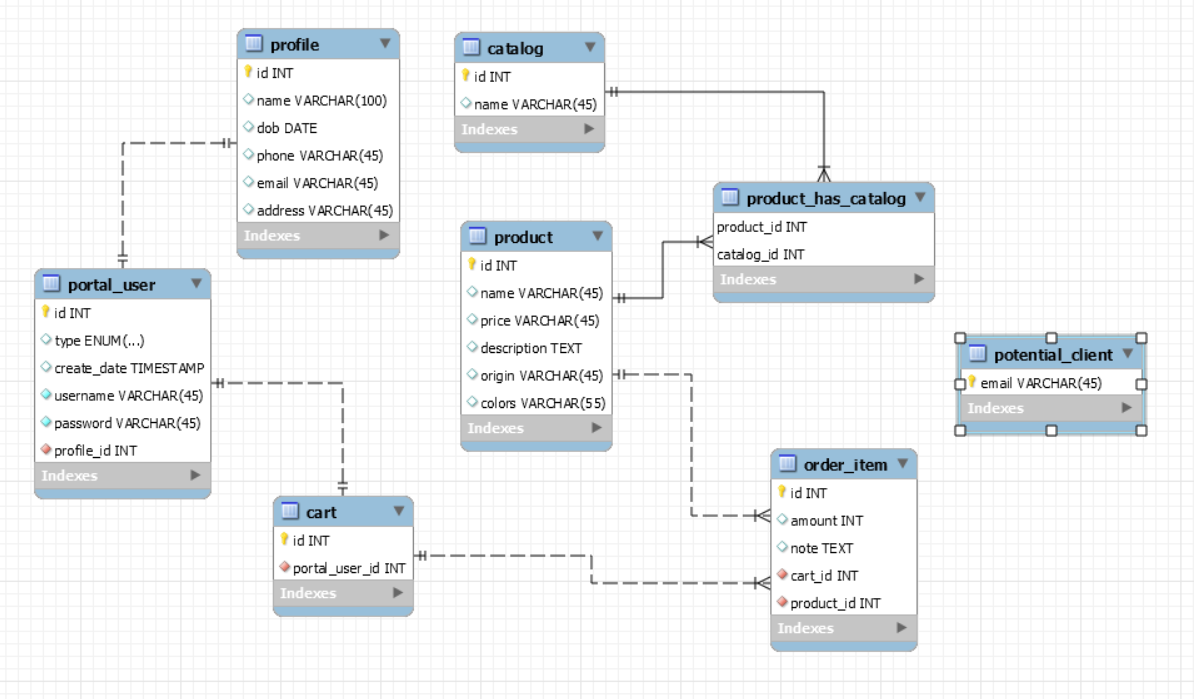


* ER Diagram



## Task 3

* Database dictionary – Physical Design( Database Diagram)



## Task 4

* Implement database in MySQL

drop database if exists portal;

CREATE database portal;

use portal;

CREATE TABLE IF NOT EXISTS `portal`.`profile` (

`id` INT NOT NULL AUTO\_INCREMENT,

`name` VARCHAR(100) NULL,

`dob` DATE NULL,

`phone` VARCHAR(45) NULL,

`email` VARCHAR(45) NULL,

`address` VARCHAR(45) NULL,

PRIMARY KEY (`id`))

ENGINE = InnoDB;

CREATE TABLE IF NOT EXISTS `portal`.`portal\_user` (

`id` INT NOT NULL AUTO\_INCREMENT,

`type` ENUM('SELLER', 'CONSUMER', 'ADMIN') NULL,

`create\_date` TIMESTAMP NULL,

`username` VARCHAR(45) NOT NULL,

`password` VARCHAR(45) NOT NULL,

`profile\_id` INT NOT NULL,

PRIMARY KEY (`id`),

UNIQUE INDEX `username\_UNIQUE` (`username` ASC) VISIBLE,

INDEX `fk\_portal\_user\_profile\_idx` (`profile\_id` ASC) VISIBLE,

CONSTRAINT `fk\_portal\_user\_profile`

FOREIGN KEY (`profile\_id`)

REFERENCES `portal`.`profile` (`id`)

ON DELETE NO ACTION

ON UPDATE NO ACTION);

CREATE TABLE IF NOT EXISTS `portal`.`catalog` (

`id` INT NOT NULL AUTO\_INCREMENT,

`name` VARCHAR(45) NULL,

PRIMARY KEY (`id`));

CREATE TABLE IF NOT EXISTS `portal`.`product` (

`id` INT NOT NULL AUTO\_INCREMENT,

`name` VARCHAR(45) NULL,

`price` VARCHAR(45) NULL,

`description` TEXT NULL,

`origin` VARCHAR(45) NULL,

`colors` VARCHAR(55) NULL,

PRIMARY KEY (`id`));

CREATE TABLE IF NOT EXISTS `portal`.`cart` (

`id` INT NOT NULL AUTO\_INCREMENT,

`portal\_user\_id` INT NOT NULL,

PRIMARY KEY (`id`),

INDEX `fk\_cart\_portal\_user1\_idx` (`portal\_user\_id` ASC) VISIBLE,

CONSTRAINT `fk\_cart\_portal\_user1`

FOREIGN KEY (`portal\_user\_id`)

REFERENCES `portal`.`portal\_user` (`id`)

ON DELETE NO ACTION

ON UPDATE NO ACTION);

CREATE TABLE IF NOT EXISTS `portal`.`potential\_client` (

`email` VARCHAR(45) NOT NULL,

PRIMARY KEY (`email`));

CREATE TABLE IF NOT EXISTS `portal`.`potential\_client` (

`email` VARCHAR(45) NOT NULL,

PRIMARY KEY (`email`));

CREATE TABLE IF NOT EXISTS `portal`.`order\_item` (

`id` INT NOT NULL AUTO\_INCREMENT,

`amount` INT NULL,

`note` TEXT NULL,

`cart\_id` INT NOT NULL,

`product\_id` INT NOT NULL,

PRIMARY KEY (`id`),

INDEX `fk\_order\_item\_cart1\_idx` (`cart\_id` ASC) VISIBLE,

INDEX `fk\_order\_item\_product1\_idx` (`product\_id` ASC) VISIBLE,

CONSTRAINT `fk\_order\_item\_cart1`

FOREIGN KEY (`cart\_id`)

REFERENCES `portal`.`cart` (`id`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_order\_item\_product1`

FOREIGN KEY (`product\_id`)

REFERENCES `portal`.`product` (`id`)

ON DELETE NO ACTION

ON UPDATE NO ACTION);

CREATE TABLE IF NOT EXISTS `portal`.`product\_catalog` (

`product\_id` INT NOT NULL,

`catalog\_id` INT NOT NULL,

PRIMARY KEY (`product\_id`, `catalog\_id`),

INDEX `fk\_product\_has\_catalog\_catalog1\_idx` (`catalog\_id` ASC) VISIBLE,

INDEX `fk\_product\_has\_catalog\_product1\_idx` (`product\_id` ASC) VISIBLE,

CONSTRAINT `fk\_product\_has\_catalog\_product1`

FOREIGN KEY (`product\_id`)

REFERENCES `portal`.`product` (`id`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_product\_has\_catalog\_catalog1`

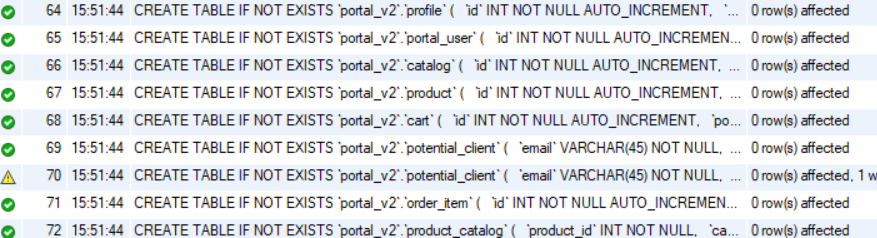
FOREIGN KEY (`catalog\_id`)

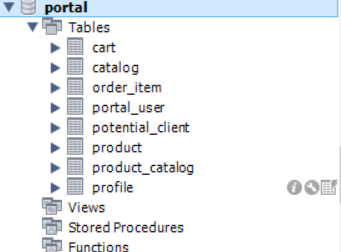
REFERENCES `portal`.`catalog` (`id`)

ON DELETE NO ACTION

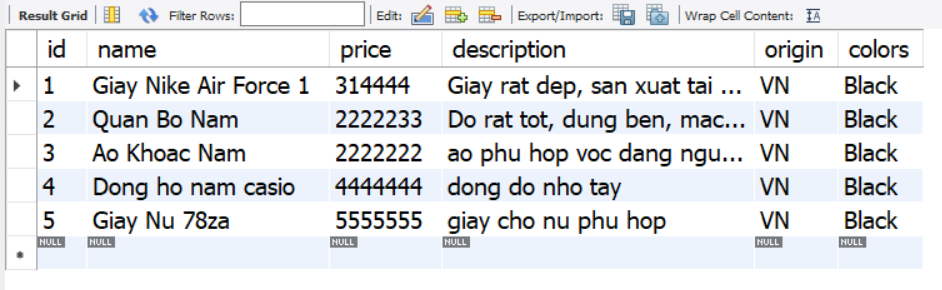
ON UPDATE NO ACTION);

* Produce the Screen capture of created tables

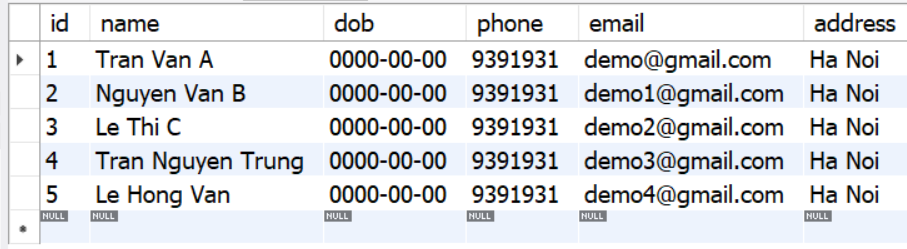




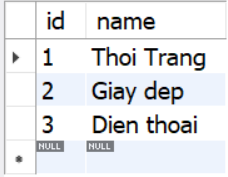
* + Table product



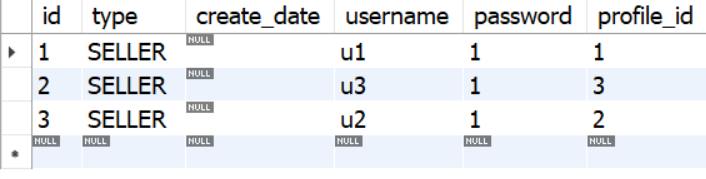
* + Table profile



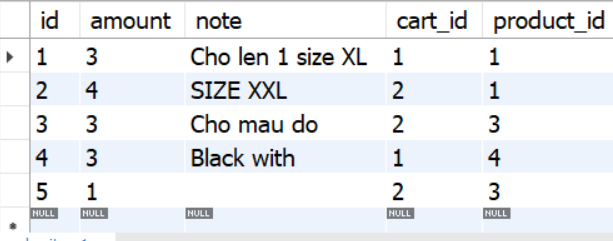
* + Table catalog



* + Table portal\_user



* + Table order\_item



## Task 5

* Create index
  + Script

CREATE INDEX profile\_email\_idx on profile(email);

CREATE INDEX profile\_phone\_idx on profile(phone);

CREATE INDEX profile\_dob\_idx on profile(dob);

CREATE INDEX portaluser\_username\_idx on portal\_user(username);

CREATE INDEX product\_price\_idx on product(price);

* + Screenshot of index

|  |  |  |
| --- | --- | --- |
| **Table** | **Index name** | **Query** |
| profile | profile\_email\_idx  profile\_phone\_idx  profile\_dob\_idx | CREATE INDEX profile\_email\_idx on profile(email);  CREATE INDEX profile\_phone\_idx on profile(phone);  CREATE INDEX profile\_dob\_idx on profile(dob); |
| Portal\_user | portaluser\_username\_idx | CREATE INDEX portaluser\_username\_idx on portal\_user(username); |
| Product | product\_price\_idx | CREATE INDEX product\_price\_idx on product(price); |

* + Rationale for index creation

|  |  |
| --- | --- |
| **Index name** | **Retionale** |
| profile\_email\_idx | Helps search users by email more effectively.  SELECT \* FROM profile where email |
| profile\_phone\_idx | Helps search users by phone more effectively.  SELECT \* FROM profile where phone = ? |
| profile\_dob\_idx | Helps search users by date of birth more effectively.  SELECT \* FROM profile where dob = ? |
| portaluser\_username\_idx | Helps search users by username more effectively when login.  SELECT \* FROM portal\_user where username = ? |
| product\_price\_idx | Helps search users by price in range more effectively.  SELECT \* FROM portal\_user where price >= ? and price <= ? |

* Create a Backup script to back up the database every 6 hours.
  + Script

@echo off

setlocal

REM Set MySQL credentials

set MYSQL\_USER=root

set MYSQL\_PASSWORD=your\_password

set MYSQL\_DATABASE=your\_database

REM Set backup directory

set BACKUP\_DIR=C:\Path\To\Your\Backup\Directory

REM Get current date and time

for /f "tokens=1-4 delims=/ " %%i in ('date /t') do (

set YYYY=%%l

set MM=%%j

set DD=%%k

)

for /f "tokens=1-2 delims=: " %%i in ('time /t') do (

set HH=%%i

set MIN=%%j

)

REM Create timestamp for the backup file

set TIMESTAMP=%YYYY%\_%MM%\_%DD%\_%HH%\_%MIN%

REM Set the path to mysqldump (adjust based on your MySQL installation)

set MYSQLDUMP\_PATH=C:\Path\To\Your\MySQL\bin\mysqldump.exe

REM Perform the backup

"%MYSQLDUMP\_PATH%" -u %MYSQL\_USER% -p%MYSQL\_PASSWORD% %MYSQL\_DATABASE% > "%BACKUP\_DIR%\backup\_%TIMESTAMP%.sql"

echo Backup completed successfully at %TIMESTAMP%

endlocal

## Task 6

* Create 8 SQL queries which will be used by the Community portal

|  |  |  |
| --- | --- | --- |
| **Page** | **Purpose of query** | **Query** |
| Registration | For account creation of new users | INSERT INTO portal\_user (`type`, username, `password`, profile\_id) VALUES ('ADMIN', 'admin', '123', 1); |
| Update profile info | For update profile info | UPDATE `profile` SET `name` = 'Tran Van A', phone='039193910', email='demo@gmail.com' where id = 1; |
| Add New Product | Add new product | INSERT INTO product(name, price, description, origin, colors)  VALUES('Giay nike air force 1', 300000, 'GIay dep', 'VIETNAM', 'WHITE, BLACK'); |
| Search Product | Search product by name | SELECT \* FROM PRODUCT where name like '%air force 1%'; |
| Get product info | Get product info by id | SELECT \* FROM PRODUCT where id = 1; |
| Update product info | Update product info | UPDATE product set price=90000, origin='CHINA' where id = 1; |
| Add new catalog | Add new catalog | INSERT INTO catalog(name) values ('Thoi trang'), ('Giay dep'); |
| Add product to cart | Add product to cart | INSERT INTO order\_item(amount, note, cart\_id, product\_id) values(3, 'None', 2, 2); |

* Query Report

|  |  |
| --- | --- |
| **Title of the report** | Total amount sell of one product |
| **Description** | Get the total sales amount of a product |
| **Purpose** | Get the total sales amount of a product |
| **Query** | select sum(amount) as TotalAmountSeller from order\_item where product\_id = '1' |
| **Result screenshot** |  |

|  |  |
| --- | --- |
| **Title of the report** | Total revenue sell of one product |
| **Description** | Get the total revenue of a product |
| **Purpose** | Get the total revenue of a product |
| **Query** | select sum(amount \* p.price) as TotalRevenue from order\_item inner join product as p  on p.id = order\_item.product\_id where product\_id = '1' |
| **Result screenshot** |  |

|  |  |
| --- | --- |
| **Title of the report** | Calculate the total sales revenue of a consumer |
| **Description** | Statistic the total sales revenue of a consumer |
| **Purpose** | the total sales revenue of a consumer |
| **Query** | select profile.name, profile.phone, profile.email, sum(amount \* p.price) as TotalRevenue from order\_item inner join product as p  on p.id = order\_item.product\_id inner join cart on cart.id = order\_item.cart\_id  inner join portal\_user on portal\_user.id = cart.portal\_user\_id inner join profile  on profile.id = portal\_user.profile\_id  group by cart.portal\_user\_id  order by TotalRevenue desc |
| **Result screenshot** |  |

* Import Sample data from CSV file in to the database
  + Table catalog
* Command

LOAD DATA local INFILE 'C:/Users/ADMIN/Documents/dbms\_res/catalog.csv'

INTO TABLE portal.catalog

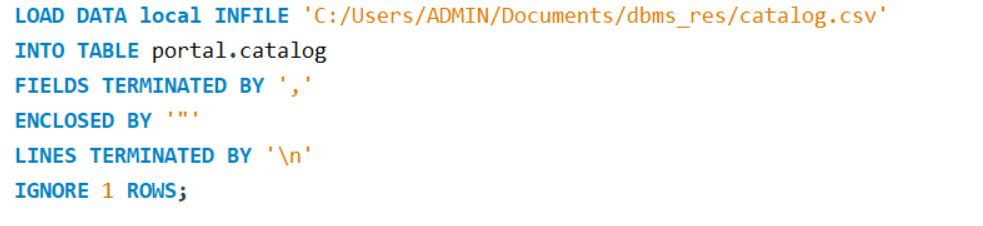
FIELDS TERMINATED BY ','

ENCLOSED BY '"'

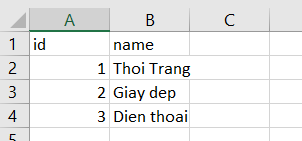
LINES TERMINATED BY '\n'

IGNORE 1 ROWS;

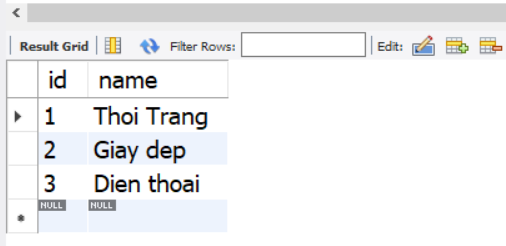
* Command



* Csv file



* SQL Result



* + Table profile
* Command

LOAD DATA local INFILE 'C:/Users/ADMIN/Documents/dbms\_res/profile.csv'

INTO TABLE portal.profile

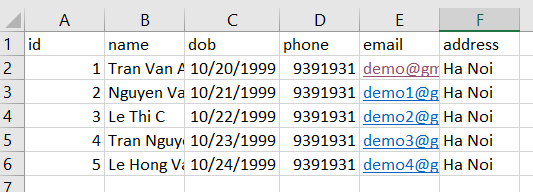
FIELDS TERMINATED BY ','

ENCLOSED BY '"'

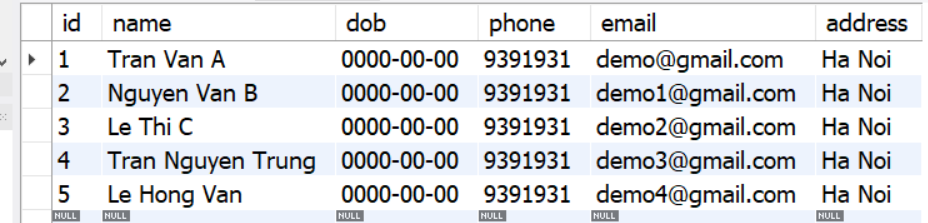
LINES TERMINATED BY '\n'

IGNORE 1 ROWS;

* Csv file



* SQL Result



* + Table product
* Command

LOAD DATA local INFILE 'C:/Users/ADMIN/Documents/dbms\_res/product.csv'

INTO TABLE portal.product

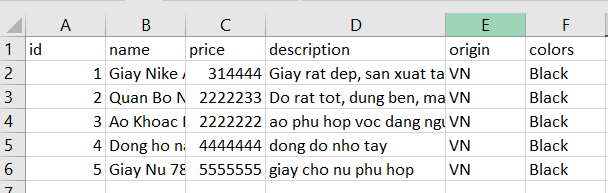
FIELDS TERMINATED BY ','

ENCLOSED BY '"'

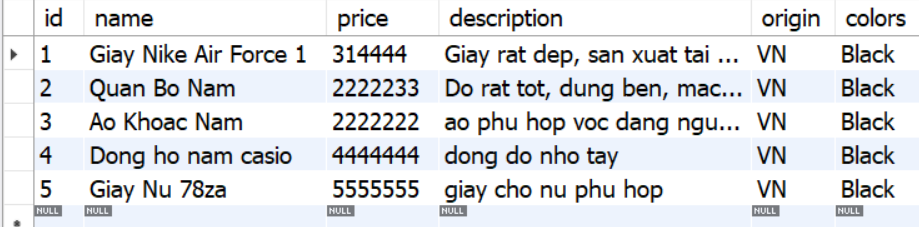
LINES TERMINATED BY '\n'

IGNORE 1 ROWS;

* Csv file



* SQL Result



* + Table order\_item
* Command

LOAD DATA local INFILE 'C:/Users/ADMIN/Documents/dbms\_res/order\_item.csv'

INTO TABLE portal.order\_item

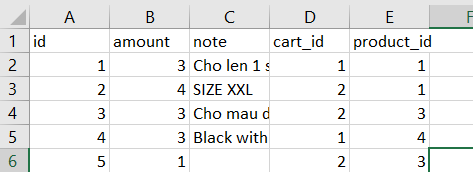
FIELDS TERMINATED BY ','

ENCLOSED BY '"'

LINES TERMINATED BY '\n'

IGNORE 1 ROWS;

* Csv file



* SQL Result

