

TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN TOÁN ỨNG DỤNG VÀ TIN HỌC

—oo—



BÁO CÁO CUỐI KỲ

TRÌNH BÀY NỘI DUNG THỰC HÀNH HÀNG
TUẦN

Bộ môn: Cơ sở dữ liệu

Giảng viên hướng dẫn: Ths.Nguyễn Danh Tú

Sinh viên: Nguyễn Công Hiếu - 20195016

Lớp: Toán - tin 02

HÀ NỘI, 8/2021

Lời nói đầu

Trong thời đại của công nghệ số, nơi mà thông tin, tri thức được biểu diễn qua những con số từ đó dẫn tới một khối lượng lớn những dữ liệu cần được xử lý và lưu trữ một cách khoa học. Để giải quyết những vấn đề đó trong thực tiễn kinh tế - kỹ thuật - xã hội, các *Cơ Sở Dữ Liệu* đã được thiết kế, áp dụng để chúng ta có thể tiếp cận tới thông tin một cách dễ dàng.

Thông qua 16 tiết của bộ môn *Cơ Sở Dữ Liệu* và sự hướng dẫn tận tình của thầy **Nguyễn Danh Tú**, em đã phần nào nắm được những kiến thức căn bản nhất của một CSDL như cách thiết kế, vận hành, sửa đổi,... cũng như cách để sử dụng *Hệ Cơ Sở Dữ Liệu*.

Em xin chân thành cảm ơn những kiến thức quý báu mà thầy đã truyền dạy!

Mục lục

Chương 1

Nội dung các tuần

1.1 Tuần 1

1.1.1 Bài 1

Thiết lập hệ quản trị cơ sở dữ liệu trên máy tính cá nhân.

INPUT: Click vào đường link sau đây và chọn tải bản phù hợp với máy tính.

OUTPUT: Sau khi hoàn các bước được hướng dẫn khi cài MySQL, ta có thể kiểm tra bằng một số lệnh thông qua command line.

SOURCE CODE: `mysqladmin -u root -p version`

DISPLAY RESULTS:

```
C:\Program Files\MySQL\MySQL Server 8.0\bin>mysqladmin -u root -p version
Enter password: *****
mysqladmin Ver 8.0.23 for Win64 on x86_64 (MySQL Community Server - GPL)
Copyright (c) 2000, 2021, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Server version      8.0.23
Protocol version   10
Connection          localhost via TCP/IP
TCP port            3306
Uptime:             2 hours 17 min 42 sec

Threads: 2  Questions: 4  Slow queries: 0  Opens: 117  Flush tables: 3  Open tables: 36  Queries per second avg: 0.000
```

1.1.2 Bài 2

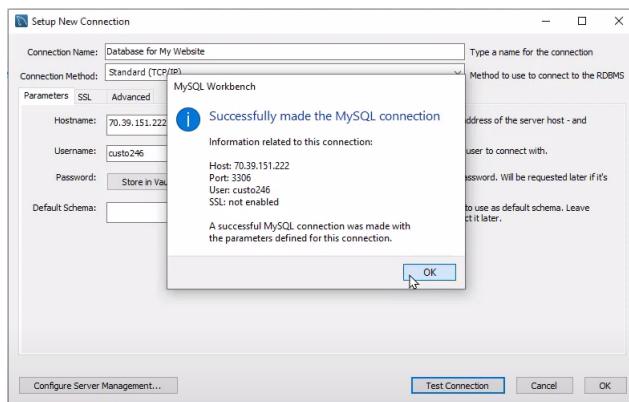
Thiết lập được môi trường làm việc với cơ sở dữ liệu từ máy chủ đơn vị.

INPUT:

- Host name
- Port
- User
- Password

ALGORITHM: Thông qua MySQL Workbench ta có thể truy cập vào các server khác nhau.

DISPLAY RESULTS:



1.1.3 Bài 3

Tạo một cơ sở dữ liệu mẫu để làm việc.

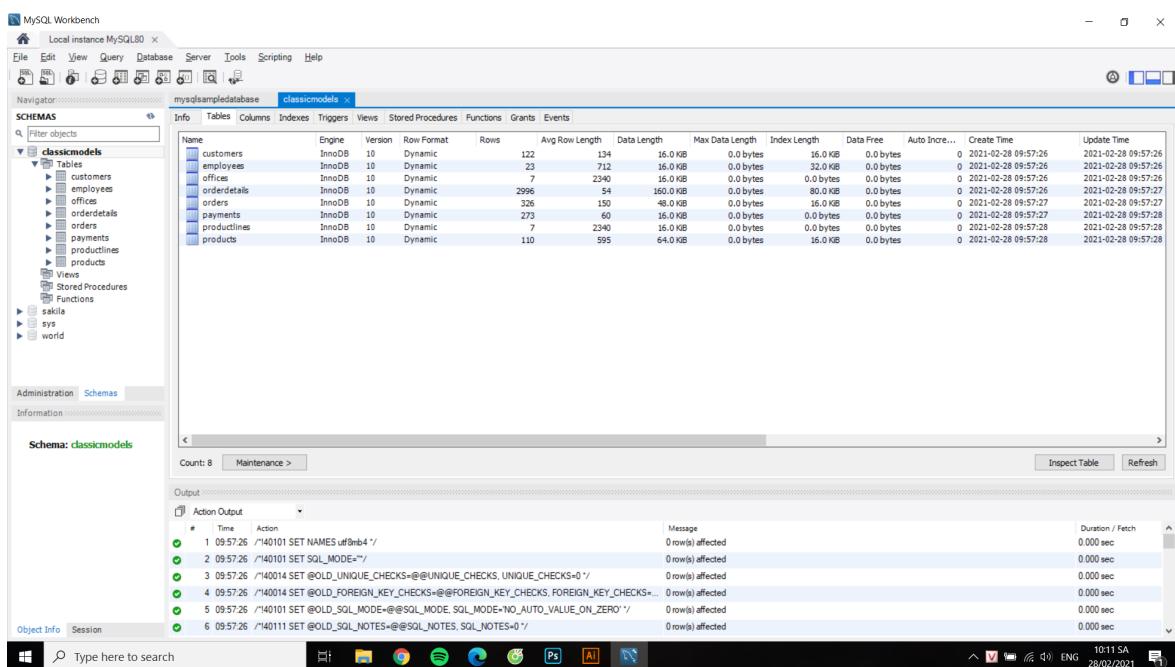
INPUT: Click vào link này để lấy sample DB.

OUTPUT: Ta sẽ được một CSDL lưu trữ trong hệ quản trị CSDL.

ALGORITHM:

- Qua Export, Import:
 - Cách 1: Trong MySQL Workbench trên thanh công cụ \Rightarrow Server \Rightarrow Data import/Data Export.
 - Cách 2: Vào command line gõ `mysql -u username -p databaseName < file.sql`
- Qua Copy, Paste: Mở file .sql \Rightarrow Ctrl-C \Rightarrow Tạo 1 script mới và Ctrl-V.
- Qua Backup, Restore: Trong command line, gõ `mysql -u [user] -p [database-Name] < [filename].sql` nếu đã tồn tại một database được backup trước đó.
- Qua Attach, Detach:

DISPLAY RESULTS: Sau khi thực hiện ta được một cơ sở dữ liệu mẫu.



1.2 Tuần 2

1.2.1 Bài 1

SELECT, ORDER BY, WHERE, SELECT DISTINCT

INPUT:

OUTPUT:

SOURCE CODE:

```
SELECT DISTINCT salesRepEmployeeNumber, city, country
FROM customers
WHERE state IS NOT NULL
ORDER BY customerName;
```

ALGORITHM: Lấy các thông tin(unique) salesRepEmployeeNumber, city, country từ bảng customers với điều kiện state khác NULL và sắp xếp theo customerName.

DISPLAY RESULTS:

| | salesRepEmployeeNumber | city | country |
|------|------------------------|----------------|-----------|
| ▶ | 1286 | New Haven | USA |
| | 1611 | North Sydney | Australia |
| NULL | | Cork | Ireland |
| | 1611 | Glen Waverly | Australia |
| | 1611 | Melbourne | Australia |
| | 1611 | South Brisbane | Australia |
| | 1216 | Brickhaven | USA |
| | 1166 | Glendale | USA |
| | 1188 | Cambridge | USA |
| | 1323 | Vancouver | Canada |
| | 1188 | Philadelphia | USA |
| | 1166 | San Diego | USA |
| | 1216 | Allentown | USA |
| | 1188 | Boston | USA |
| | 1216 | New Bedford | USA |
| | 1323 | Bridgewater | USA |
| | 1323 | Glendale | USA |
| | 1216 | Boston | USA |
| | 1501 | Cowes | UK |
| | 1323 | NYC | USA |
| | ... | ... | ... |

1.2.2 Bài 2

AND, OR, IN

INPUT:

OUTPUT:

SOURCE CODE:

```
SELECT * FROM orderdetails
WHERE (orderLineNumber = 1 AND priceEach > 100.00)
OR (quantityOrdered IN (20,30,40));
```

ALGORITHM: Lấy tất cả các thông tin từ bảng orderdetails với điều kiện hoặc là quantityOrdered nằm trong bộ (20,30,40) hoặc là xảy ra đồng thời orderLineNumber = 1 và priceEach > 100.00

DISPLAY RESULTS:

| | orderNumber | productCode | quantityOrdered | priceEach | orderLineNumber |
|---|-------------|-------------|-----------------|-----------|-----------------|
| ▶ | 10100 | S18_1749 | 30 | 136.00 | 3 |
| | 10101 | S18_2795 | 26 | 157.06 | 1 |
| | 10103 | S24_2300 | 36 | 107.34 | 1 |
| | 10104 | S12_3148 | 34 | 131.44 | 1 |
| | 10106 | S700_2047 | 30 | 85.09 | 16 |
| | 10106 | S700_2834 | 32 | 113.90 | 1 |
| | 10107 | S10_1678 | 30 | 81.35 | 2 |
| | 10107 | S12_2823 | 21 | 122.00 | 1 |
| | 10107 | S32_1374 | 20 | 88.90 | 8 |
| | 10108 | S24_3371 | 30 | 60.01 | 5 |
| | 10108 | S24_3856 | 40 | 132.00 | 1 |
| | 10109 | S18_2870 | 26 | 126.72 | 1 |
| | 10110 | S18_2795 | 31 | 163.69 | 1 |

1.2.3 Bài 3

BETWEEN, LIKE, LIMIT

INPUT:

OUTPUT:

SOURCE CODE:

```
SELECT customerNumber, salesRepEmployeeNumber
FROM customers
WHERE (salesRepEmployeeNumber LIKE '11__')
AND (customerNumber BETWEEN 200 AND 300)
ORDER BY customerNumber
LIMIT 2,2;
```

ALGORITHM: Lấy các thông tin customerNumber, salesRepEmployeeNumber từ bảng customers với điều kiện đồng thời xảy ra salesRepEmployeeNumber có dạng 11__ và customerNumber thuộc khoảng (200;300). Các thông tin được sắp xếp theo customerNumber và chỉ lấy ra 2 bộ dữ liệu bắt đầu từ bộ thứ 3.

DISPLAY RESULTS:

| | customerNumber | salesRepEmployeeNumber |
|---|----------------|------------------------|
| ▶ | 219 | 1166 |
| | 239 | 1166 |
| * | NULL | NULL |

1.2.4 Bài 4

IS NULL, TABLE/COLUMN ALIASES

INPUT:

OUTPUT:

SOURCE CODE:

```

SELECT
    customerNumber AS id,
    salesRepEmployeeNumber AS SREN
FROM customers AS table_1
WHERE (salesRepEmployeeNumber IS NULL);

```

ALGORITHM: Chọn các thông tin như customerNumber(rút gọn thành id), salesRepEmployeeNumber(rút gọn thành SREN) từ bảng customers(rút gọn thành table_1 với điều kiện salesRepEmployeeNumber là NULL).

DISPLAY RESULTS:

| | id | SREN |
|---|-----|------|
| ▶ | 125 | NULL |
| | 169 | NULL |
| | 206 | NULL |
| | 223 | NULL |
| | 237 | NULL |
| | 247 | NULL |
| | 273 | NULL |
| | 293 | NULL |
| | 303 | NULL |
| | 397 | NULL |

customers 6 ×

1.2.5 Bài 5

INNER JOIN, LEFT JOIN, RIGHT JOIN, SELF JOIN, CROSS JOIN

INPUT:

OUTPUT:

SOURCE CODE:

- INNER JOIN

```

SELECT c.customerNumber, c.contactFirstName, o.orderDate
FROM customers AS c
INNER JOIN orders AS o
ON c.customerNumber = o.customerNumber;

```

- LEFT JOIN

```

SELECT c.customerNumber, c.contactFirstName, o.orderDate
FROM customers AS c
LEFT JOIN orders AS o USING(customerNumber);

```

- RIGHT JOIN

```

SELECT e.officeCode, e.firstName, e.jobTitle
FROM offices AS o
RIGHT JOIN employees AS e USING(officeCode);

```

- SELF JOIN

```
SELECT e1.employeeNumber,
       e1.firstName AS employeeName,
       e2.firstName AS managerName
  FROM employees AS e1
 LEFT JOIN employees AS e2
    ON e2.employeeNumber = e1.reportsTo;
```

- CROSS JOIN

```
SELECT o1.officeCode, o2.city
  FROM offices AS o1
CROSS JOIN offices AS o2
 ORDER BY officeCode;
```

ALGORITHM:

- INNER JOIN: Lấy ra từ bảng customer inner join với bảng orders các thông tin như customerNumber, contactFirstName, orderDate với điều kiện join là cột customerNumber.
- LEFT JOIN: Giống như INNER JOIN ở phía trên nhưng thay bằng LEFT JOIN
- RIGHT JOIN: Liên kết bên phải của bảng offices với bảng employees trên cột officeCode rồi lấy ra các thông tin officeCode, firstName và jobTitle.
- SELF JOIN: Lấy ra danh sách nhân viên - quản lý tương ứng bằng cách dùng bảng employees join với chính nó dựa trên 2 cột là employeeNumber và reportsTo.
- CROSS JOIN: Lấy ra các thông tin như officeCode và city từ việc sử dụng tích descart của bảng offices với chính nó và sắp xếp theo officeCode.

DISPLAY RESULTS:

- INNER JOIN

| | customerNumber | contactFirstName | orderDate |
|------------|----------------|------------------|------------|
| ▶ | 363 | Dorothy | 2003-01-06 |
| | 128 | Roland | 2003-01-09 |
| | 181 | Michael | 2003-01-10 |
| | 121 | Jonas | 2003-01-29 |
| | 141 | Diego | 2003-01-31 |
| | 145 | Jytte | 2003-02-11 |
| | 278 | Giovanni | 2003-02-17 |
| | 131 | Kwai | 2003-02-24 |
| | 385 | Arnold | 2003-03-03 |
| | 486 | Rosa | 2003-03-10 |
| | 187 | Rachel | 2003-03-18 |
| Result 1 × | | | |

- LEFT JOIN

| | customerNumber | contactFirstName | orderDate |
|------------|----------------|------------------|------------|
| ▶ | 103 | Carine | 2003-05-20 |
| | 103 | Carine | 2004-09-27 |
| | 103 | Carine | 2004-11-25 |
| | 112 | Jean | 2003-05-21 |
| | 112 | Jean | 2004-08-06 |
| | 112 | Jean | 2004-11-29 |
| | 114 | Peter | 2003-04-29 |
| | 114 | Peter | 2003-05-21 |
| | 114 | Peter | 2004-02-20 |
| | 114 | Peter | 2004-11-24 |
| | 114 | Peter | 2004-11-29 |
| Result 2 × | | | |

- RIGHT JOIN

| | officeCode | firstName | jobTitle |
|------------|------------|-----------|----------------------|
| ▶ | 1 | Diane | President |
| | 1 | Mary | VP Sales |
| | 1 | Jeff | VP Marketing |
| | 6 | William | Sales Manager (APAC) |
| | 4 | Gerard | Sale Manager (EMEA) |
| | 1 | Anthony | Sales Manager (NA) |
| | 1 | Leslie | Sales Rep |
| | 1 | Leslie | Sales Rep |
| | 2 | Julie | Sales Rep |
| | 2 | Steve | Sales Rep |
| | 3 | Foon Yue | Sales Rep |
| Result 3 × | | | |

- SELF JOIN

| | employeeNumber | employeeName | managerName |
|------------|----------------|--------------|-------------|
| ▶ | 1002 | Diane | NULL |
| | 1056 | Mary | Diane |
| | 1076 | Jeff | Diane |
| | 1088 | William | Mary |
| | 1102 | Gerard | Mary |
| | 1143 | Anthony | Mary |
| | 1165 | Leslie | Anthony |
| | 1166 | Leslie | Anthony |
| | 1188 | Julie | Anthony |
| | 1216 | Steve | Anthony |
| | 1286 | Foon Yue | Anthony |
| Result 4 × | | | |

- CROSS JOIN

| | officeCode | city |
|------------|------------|---------------|
| ▶ | 1 | San Francisco |
| | 1 | Paris |
| | 1 | Sydney |
| | 1 | London |
| | 1 | Boston |
| | 1 | Tokyo |
| | 1 | NYC |
| | 2 | London |
| | 2 | San Francisco |
| | 2 | Boston |
| | 2 | Sydney |
| Result 5 × | | |

1.2.6 Bài 6

GROUP BY, HAVING, ROLLUP

INPUT:

OUTPUT:

SOURCE CODE:

```

SELECT DISTINCT
    p.productVendor AS brand
    p.productLine AS category,
    SUM(quantityOrdered * priceEach) AS sales
FROM products AS p
INNER JOIN orderdetails AS o USING(productCode)
GROUP BY
    brand, category WITH ROLLUP
HAVING sales < 100000;

```

ALGORITHM: Từ bảng products join với bảng orderdetails dựa trên cột productCode, lấy ra productVendor(brand), productLine(category) và doanh số ($sales = quantityOrdered \cdot priceEach$). Tất cả được nhóm theo (brand, category) với điều kiện sales < 100000.

DISPLAY RESULTS:

| | brand | category | sales |
|---|--------------------------|------------------|----------|
| ▶ | Autoart Studio Design | Planes | 58434.07 |
| | Autoart Studio Design | Ships | 56455.11 |
| | Carousel DieCast Legends | Trains | 82617.12 |
| | Carousel DieCast Legends | Trucks and Buses | 55835.30 |
| | Classic Metal Creations | Planes | 94885.37 |
| | Classic Metal Creations | Trains | 52123.81 |
| | Classic Metal Creations | Trucks and Buses | 46519.05 |
| | Classic Metal Creations | Vintage Cars | 38449.09 |
| | Exoto Designs | Motorcycles | 89364.89 |
| | Gearbox Collectibles | Motorcycles | 33268.76 |
| | Gearbox Collectibles | Planes | 68741.91 |

Result 6 ×

1.2.7 Bài 7

SUBQUERY, DERIVED TABLES, EXISTS

INPUT: Kiểm tra xem khách hàng nào có vấn đề trong việc giao nhận hàng.

OUTPUT: Đưa ra bảng gồm id khách hàng, cột kết quả.

SOURCE CODE:

```

SELECT
    c.customerNumber,
    (CASE
        WHEN o.comments IS NULL THEN 'NO'
        WHEN o.comments IS NOT NULL THEN 'YES'
    END) AS problems
FROM customers AS c
JOIN orders AS o USING(customerNumber)
WHERE EXISTS (
    SELECT 1
    FROM orders
    GROUP BY customerNumber
    HAVING COUNT(comments IS NOT NULL) > 4
    ORDER BY customerNumber;
)

```

ALGORITHM:

- Bước 1: Tạo một derived table

Từ bảng orders, nhóm theo customerNumber với điều kiện tổng các comments khác NULL lớn hơn 4.

- Bước 2: Từ bảng customers join với orders trên cột customerNumber, lấy ra customerNumber và cột problems('YES' nếu comment là NULL, 'NO' nếu comment khác NULL). Với điều kiện là hàng chứa customerNumber phải chứa trong derived table ở trên.

DISPLAY RESULTS:

| | customerNumber | problems |
|---|----------------|----------|
| ▶ | 103 | NO |
| | 103 | NO |
| | 103 | NO |
| | 112 | YES |
| | 112 | NO |
| | 112 | NO |
| | 114 | YES |

1.2.8 Bài 8

UNION, MINUS, INTERSECT

INPUT:

- UNION: Lấy ra tập hợp các giá trị id của bảng customers và bảng orders.
- MINUS: Lấy ra tập hợp các giá trị id của bảng customers sao cho creditLimit nhỏ hơn 100000.
- INTERSECT: Lấy ra tập khách hàng đã thực hiện thanh toán.

OUTPUT:

- UNION: Một bảng với cột customerNumber.
- MINUS: Một bảng với cột customerNumber.
- INTERSECT: Một bảng với cột customerNumber.

SOURCE CODE:

- UNION

```
SELECT customerNumber FROM customers
UNION
SELECT customerNumber FROM orders;
```

- MINUS

```
SELECT c1.customerNumber
FROM customers AS c1
LEFT JOIN customers AS c2 USING(customerNumber)
WHERE c1.creditLimit < 100000
ORDER BY customerNumber;
```

- INTERSECT

```
SELECT DISTINCT customerNumber
FROM customers
JOIN payments USING(customerNumber);
```

ALGORITHM:

- UNION: Dùng toán tử UNION cho 2 tập customerNumber từ bảng customer và orders.
- MINUS: Dùng left join trên customers vào chính nó dựa trên cột customerNumber với điều kiện creditLimit < 100000.
- INTERSECT: Chọn phân biệt(DISTINCT) customerNumber từ bảng customers join với payments sử dụng cột customerNumber

DISPLAY RESULTS:

- UNION

| | customerNumber |
|---|----------------|
| ▶ | 125 |
| | 169 |
| | 206 |
| | 223 |
| | 237 |
| | 247 |
| | 273 |
| | 293 |
| | 303 |
| | 307 |
| | 335 |

Result 9 ×

- MINUS

| customerNumber |
|----------------|
| 103 |
| 112 |
| 121 |
| 125 |
| 128 |
| 129 |
| 144 |
| 145 |
| 161 |
| 166 |
| 167 |

- INTERSECT

| customerNumber |
|----------------|
| 103 |
| 112 |
| 114 |
| 119 |
| 121 |
| 124 |
| 128 |
| 129 |
| 131 |
| 141 |

1.3 Tuần 3

1.3.1 Bài 1

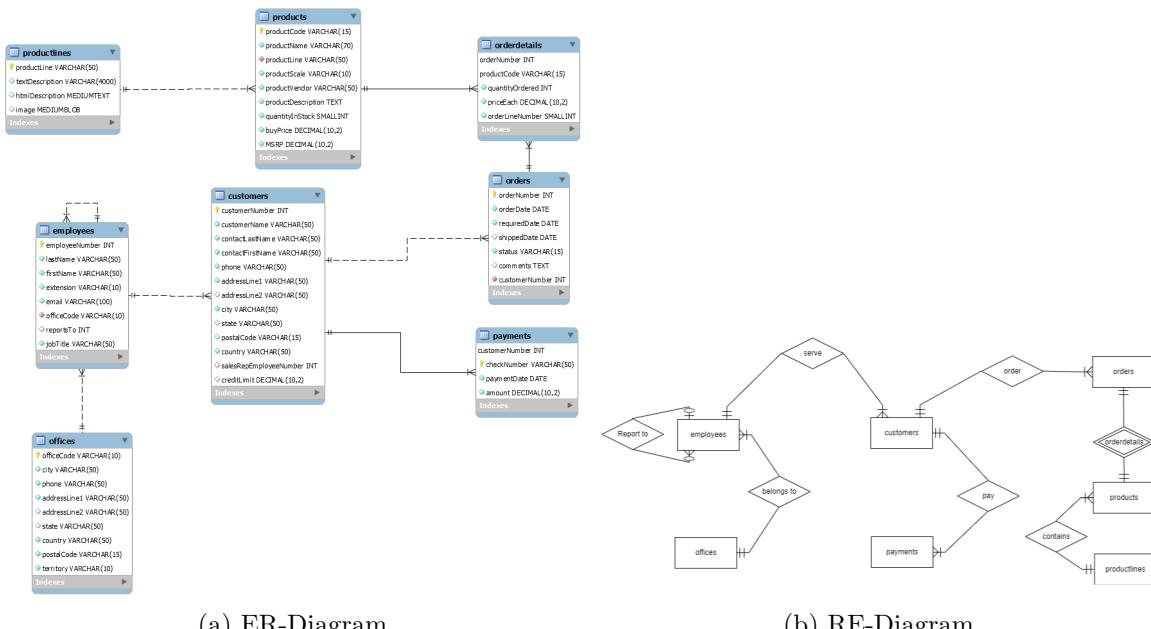
Xây dựng lược đồ (Diagram)

INPUT: Vẽ, đọc và phân tích lược đồ

OUTPUT: Hiển thị lược đồ trên MySQL Workbench, phân tích lược đồ trên file text.

ALGORITHM: Trong giao diện MySQL Workbench \Rightarrow Next cho đến khi ra được lược đồ.

DISPLAY RESULTS:



Hình 1.1: Lüct đồ Diagram

- "customers" - "employees": Với mỗi 1 hoặc nhiều khách hàng sẽ được phục vụ bởi 1 và chỉ 1 nhân viên.
Trong đó, pri_key(customers) = customerNumber, pri_key(employees) = "employeeNumber", foreign_key(employees) = "officeCode".
- "employees" - "employees": Với 1 hoặc nhiều nhân viên có thể gửi báo cáo cho 1 và chỉ một người quản lý, nhưng người lãnh đạo thì không cần gửi báo cáo cho ai.
- "employees" - "offices": 1 hoặc nhiều nhân viên có thể làm trong 1 và chỉ 1 văn phòng.
- "customers" - "payments": là mối quan hệ 1 nhiều, do khách hàng có thể trả nhiều lần.
Trong đó, pri_key(payments) = checkNumber.
- "customers" - "orders": là mối quan hệ 1 nhiều, 1 hoặc nhiều đơn hàng chỉ thuộc 1 và chỉ một khách hàng.
Với pri_key(orders) = orderNumber.
- "orders" - "products": quan hệ thông qua "orderdetails" với key là productCode, một đơn hàng ứng với một mã sản phẩm.
- "products" - "productlines": Quan hệ nhiều 1, nhiều sản phẩm thuộc 1 dòng sản phẩm.

Hình 1.2: Phân tích lược đồ

1.3.2 Bài 2

Truy vấn các bảng hệ thống trong CSDL

INPUT: Truy vấn tên các bảng, các cột của một bảng và tính chất các cột đó.

OUTPUT: Hiển thị các bảng hệ thống.

SOURCE CODE:

- Truy vấn các bảng hệ thống.

```
SELECT * FROM information_schema.tables
WHERE table_schema = 'classicmodels';
```

- Truy vấn các cột hệ thống.

```
SELECT * FROM information_schema.columns
WHERE table_schema = 'classicmodels'
ORDER BY table_name, ordinal_position;
```

DISPLAY RESULTS:

| | TABLE_CATALOG | TABLE_SCHEMA | TABLE_NAME | TABLE_TYPE | ENGINE | VERSION | ROW_FORMAT | TABLE_ROWS | AVG_ROW_LENGTH | DATA_LENGTH |
|---|---------------|---------------|--------------|------------|--------|---------|------------|------------|----------------|-------------|
| ▶ | def | classicmodels | customers | BASE TABLE | InnoDB | 10 | Dynamic | 122 | 134 | 16384 |
| | def | classicmodels | employees | BASE TABLE | InnoDB | 10 | Dynamic | 23 | 712 | 16384 |
| | def | classicmodels | offices | BASE TABLE | InnoDB | 10 | Dynamic | 7 | 2340 | 16384 |
| | def | classicmodels | orderdetails | BASE TABLE | InnoDB | 10 | Dynamic | 3008 | 54 | 163840 |
| | def | classicmodels | orders | BASE TABLE | InnoDB | 10 | Dynamic | 332 | 148 | 49152 |
| | def | classicmodels | payments | BASE TABLE | InnoDB | 10 | Dynamic | 273 | 60 | 16384 |
| | def | classicmodels | productlines | BASE TABLE | InnoDB | 10 | Dynamic | 7 | 2340 | 16384 |
| | def | classicmodels | products | BASE TABLE | InnoDB | 10 | Dynamic | 110 | 595 | 65536 |

Hình 1.3: Các bảng hệ thống

| | TABLE_CATALOG | TABLE_SCHEMA | TABLE_NAME | COLUMN_NAME | ORDINAL_POSITION | COLUMN_DEFAULT | IS_NULLABLE | DATA_TYPE |
|---|---------------|---------------|------------|---------------------------|------------------|----------------|-------------|-----------|
| ▶ | def | classicmodels | customers | customerNumber | 1 | NULL | NO | int |
| | def | classicmodels | customers | customerName | 2 | NULL | NO | varchar |
| | def | classicmodels | customers | contactLastName | 3 | NULL | NO | varchar |
| | def | classicmodels | customers | contactFirstName | 4 | NULL | NO | varchar |
| | def | classicmodels | customers | phone | 5 | NULL | NO | varchar |
| | def | classicmodels | customers | addressLine1 | 6 | NULL | NO | varchar |
| | def | classicmodels | customers | addressLine2 | 7 | NULL | YES | varchar |
| | def | classicmodels | customers | city | 8 | NULL | NO | varchar |
| | def | classicmodels | customers | state | 9 | NULL | YES | varchar |
| | def | classicmodels | customers | postalCode | 10 | NULL | YES | varchar |
| | def | classicmodels | customers | country | 11 | NULL | NO | varchar |
| | def | classicmodels | customers | salesRepEmployee... 12 | | NULL | YES | int |

Hình 1.4: Các cột hệ thống

1.4 Tuần 4

1.4.1 Bài 1 và 2

Xây dựng cấu trúc của một cơ sở dữ liệu (Data Definition)

INPUT: Thiết lập các ràng buộc(PK, UK, Datatype,...) và tạo ra các quan hệ giữa các bảng dữ liệu.

OUTPUT: Một cơ sở dữ liệu đã được định nghĩa.

SOURCE CODE:

```

CREATE DATABASE IF NOT EXISTS musicsmodel;
USE musicsmodel;
CREATE TABLE IF NOT EXISTS singers (
    singer_id INT AUTO_INCREMENT PRIMARY KEY,
    singer_name VARCHAR(255) NOT NULL,
    gender TINYTEXT NULL,
    debute_year YEAR NOT NULL,
    manager_contact VARCHAR(255),
    CONSTRAINT unique_email UNIQUE (manager_contact)
) ENGINE = INNODB;

CREATE TABLE IF NOT EXISTS songs (
    song_id INT AUTO_INCREMENT PRIMARY KEY,
    song_name VARCHAR(255) NOT NULL,
    singer_id INT NOT NULL,
    CONSTRAINT sing
    FOREIGN KEY (singer_id)
    REFERENCES singers (singer_id)
    ON UPDATE CASCADE
    ON DELETE CASCADE
) ENGINE = INNODB;

```

DISPLAY RESULTS:



1.4.2 Bài 3

Thiết lập khung nhìn (Views)

INPUT: Tạo view cho danh sách những địa chỉ email mà khách hàng có thể liên hệ.

OUTPUT: Bảng gồm id khách hàng, tên khách hàng và email của nhân viên phục vụ khách hàng.

SOURCE CODE:

```

CREATE VIEW cus_emp
AS
    SELECT
        c.customerNumber,
        c.customerName,
        e.email AS email_to
    FROM
        customers AS c
    INNER JOIN employees AS e
    WHERE e.employeeNumber = c.salesRepEmployeeNumber;

SELECT * FROM classicmodels.cus_emp;

```

ALGORITHM: Từ bảng customers join với bảng employees dựa trên cặp (employeeNumber, salesRepEmployeeNumber), lấy ra thông customerNumber, customerName, email. Rồi tạo view trên bảng đó.

DISPLAY RESULTS:

| | employeeNumber | lastName | firstName | extension | email | officeCode | reportsTo | jobTitle |
|---|----------------|-----------|-----------|-----------|---------------------------------|------------|-----------|----------------------|
| ▶ | 1002 | Murphy | Diane | x5800 | dmurphy@classicmodelcars.com | 1 | NULL | President |
| | 1056 | Patterson | Mary | x4611 | mpatterson@classicmodelcars.com | 1 | 1002 | VP Sales |
| | 1076 | Firrelli | Jeff | x9273 | jfirrelli@classicmodelcars.com | 1 | 1002 | VP Marketing |
| | 1088 | Patterson | William | x4871 | wpatterson@classicmodelcars.com | 6 | 1056 | Sales Manager (APAC) |
| | 1102 | Bondur | Gerard | x5408 | gbondur@classicmodelcars.com | 4 | 1056 | Sale Manager (EMEA) |
| | 1143 | Bow | Anthony | x5428 | abow@classicmodelcars.com | 1 | 1056 | Sales Manager (NA) |
| | 1165 | Jennings | Leslie | x3291 | ljennings@classicmodelcars.com | 1 | 1143 | Sales Rep |
| | 1166 | Thompson | Leslie | x4065 | lthompson@classicmodelcars.com | 1 | 1143 | Sales Rep |
| | 1188 | Firrelli | Julie | x2173 | jfirrelli@classicmodelcars.com | 2 | 1143 | Sales Rep |
| | 1216 | Patterson | Steve | x4334 | spatterson@classicmodelcars.com | 2 | 1143 | Sales Rep |
| | 1286 | Tseng | Foon Yue | x2248 | ftsing@classicmodelcars.com | 3 | 1143 | Sales Rep |
| | 1323 | Vanauf | George | x4102 | gvanauf@classicmodelcars.com | 3 | 1143 | Sales Rep |
| | 1337 | Bondur | Loui | x6493 | lbondur@classicmodelcars.com | 4 | 1102 | Sales Rep |

1.4.3 Bài 4

Thiết lập thủ tục và hàm(Procedures và Function)

INPUT:

- Procedure: Lấy ra tất cả bản ghi của bảng products.
- Function: Phân loại theo mức giá các sản phẩm.

OUTPUT:

- Procedure: Trả về bảng products.
- Function: Trả về bảng gồm tên sản phẩm và phân khúc sản phẩm tương ứng.

ALGORITHM:

- Procedure: Lấy tất cả từ bảng products.
- Function: Tạo hàm ProductSegment với tham số là buyPrice(số thập phân lấy 2 chữ số sau dấu phẩy). Sau đó khai báo biến product_segment kiểu varchar. Nếu buyPrice >= 100 thì product_segment = 'High', nếu 50 <= buyPrice < 100 thì product_segment = 'Middle', còn lại thì product_segment = 'Low'. Cuối cùng trả về product_segment.

SOURCE CODE:

- Procedure:

```
USE classicmodels;

DELIMITER $$

CREATE PROCEDURE GetAllProducts()

BEGIN
    SELECT * FROM products;
END $$

DELIMITER ;

CALL GetAllProducts();
```

- Function:

```
DELIMITER $$

CREATE FUNCTION ProductSegment(
    buyPrice DECIMAL(10,2)
)
RETURNS VARCHAR(255)
DETERMINISTIC
BEGIN
    DECLARE product_segment VARCHAR(255);

    IF buyPrice >= 100 THEN
```

```

        SET product_segment = 'HIGH';
ELSEIF (buyPrice >= 50 AND buyPrice < 100) THEN
    SET product_segment = 'MIDDLE';
ELSEIF buyPrice < 50 THEN
    SET product_segment = 'LOW';
END IF;

RETURN (product_segment);
END $$

DELIMITER ;
SELECT
    productName,
    ProductSegment(buyPrice)
FROM
    products
ORDER BY ProductSegment(buyPrice);

```

DISPLAY RESULTS:

- Procedure:

| productCode | productName | productLine | productScale | productVendor | productDescription | quantityInStock | buyPrice | MSRP |
|-------------|---------------------------------------|------------------|--------------|---------------------------|---|-----------------|----------|--------|
| S10_1678 | 1969 Harley Davidson Ultimate Chopper | Motorcycles | 1:10 | Min Lin Diecast | This replica features working kickstand, front su... | 7933 | 48.81 | 95.70 |
| S10_1949 | 1952 Alpine Renault 1300 | Classic Cars | 1:10 | Classic Metal Creations | Turnable front wheels; steering function; detail... | 7305 | 98.58 | 214.30 |
| S10_2016 | 1996 Moto Guzzi 1100i | Motorcycles | 1:10 | Highway 66 Mini Classics | Official Moto Guzzi logos and insignias, saddle b... | 6625 | 68.99 | 118.94 |
| S10_4698 | 2003 Harley-Davidson Eagle Drag Bike | Motorcycles | 1:10 | Red Start Diecast | Model features; official Harley Davidson logos a... | 5582 | 91.02 | 193.66 |
| S10_4757 | 1972 Alfa Romeo GTA | Classic Cars | 1:10 | Motor City Art Classics | Features include: Turnable front wheels; steeri... | 3252 | 85.68 | 136.00 |
| S10_4962 | 1962 Lancia Delta 16V | Classic Cars | 1:10 | Second Gear Diecast | Features include: Turnable front wheels; steeri... | 6791 | 103.42 | 147.74 |
| S12_1099 | 1968 Ford Mustang | Classic Cars | 1:12 | Autart Studio Design | Hood, doors and trunk all open to reveal highly ... | 68 | 95.34 | 194.57 |
| S12_1108 | 2001 Ferrari Enzo | Classic Cars | 1:12 | Second Gear Diecast | Turnable front wheels; steering function; detail... | 3619 | 95.59 | 207.80 |
| S12_1666 | 1958 Setra Bus | Trucks and Buses | 1:12 | Welly Diecast Productions | Model features 30 windows, skylights & glare re... | 1579 | 77.90 | 136.67 |
| S12_2823 | 2002 Suzuki XREO | Motorcycles | 1:12 | Unimax Art Galleries | Official logos and insignias, saddle bags located ... | 9997 | 66.27 | 150.62 |
| S12_3148 | 1969 Corvair Monza | Classic Cars | 1:18 | Welly Diecast Productions | 1:18 scale die-cast about 10" long doors open, ... | 6906 | 89.14 | 151.08 |

- Function:

| productName | ProductSegment(buyPrice) |
|--|--------------------------|
| 1962 Lancia Delta 16V | HIGH |
| 1998 Chrysler Plymouth Prowler | HIGH |
| 1969 Harley Davidson Ultimate Chopper | LOW |
| 1970 Plymouth Hemi Cuda | LOW |
| 1936 Mercedes-Benz 500K Special Roadster | LOW |
| 1911 Ford Town Car | LOW |
| 1926 Ford Fire Engine | LOW |
| P-51-D Mustang | LOW |
| 1936 Harley Davidson El Knucklehead | LOW |
| 1934 Ford V8 Coupe | LOW |
| 1969 Dodge Super Bee | LOW |

1.5 Tuần 5

1.5.1 Bài 1

INSERT, UPDATE, DELETE

INPUT Sử dụng bảng singers trong CSDL musicsmodels, thêm vào 5 bản ghi, xóa 1 bản ghi có id là 24, cập nhật lại năm sinh của ca sĩ có tên là 'Elvis Presley'.

OUTPUT: bảng singers đã được thay đổi.

SOURCE CODE:

```

USE musicsmodel;

SET SQL_SAFE_UPDATES = 0;

DELETE FROM singers
WHERE singer_id = 24;

INSERT INTO
    singers(singer_name, gender, birth_year)
VALUES
    ('Ed Sheeran', 'male', 1991),
    ('Ruel', 'male', 2002),
    ('Lukas Graham', 'male', 1988),
    ('James Arthur', 'male', 1988),
    ('Lewis Capaldi', 'male', 1996),
    ('Loran Allred', 'female', 1989),
    ('Elvis Presley', 'male', 1934);

UPDATE singers
SET
    birth_year = '1935'
WHERE singer_name = 'Elvis Presley';

SELECT * FROM singers;

```

DISPLAY RESULTS:

| | singer_id | singer_name | gender | debut_year | manager_contact |
|---|-----------|---------------|--------|------------|-----------------|
| ▶ | 1 | Ed Sheeran | male | 1991 | NULL |
| | 2 | Ruel | male | 2002 | NULL |
| | 3 | Lukas Graham | male | 1988 | NULL |
| | 4 | James Arthur | male | 1988 | NULL |
| | 5 | Lewis Capaldi | male | 1996 | NULL |
| | 6 | Loran Allred | female | 1989 | NULL |
| | 7 | Elvis Presley | male | 1935 | NULL |
| | 8 | Ed Sheeran | male | 1991 | NULL |
| | 9 | Ruel | male | 2002 | NULL |
| | 10 | Lukas Graham | male | 1988 | NULL |
| | 11 | James Arthur | male | 1988 | NULL |
| | 12 | Lewis Capaldi | male | 1996 | NULL |

singers 1 ×

1.5.2 Bài 2

Sinh tự động SQL script thông qua excel

INPUT: Dùng phần mềm excel sinh ra các câu lệnh INSERT, UPDATE, DELETE.

OUTPUT: Trả về các SQL script.

DISPLAY RESULTS:

- **INSERT:**

The screenshot shows the MySQL Workbench interface with the 'singers' table selected. The table has columns: singer_id (int AI PK), singer_name (varchar(255)), gender (tinytext), birth_year (year), and manager_contact (varchar(255)). The data grid shows rows for Low G, Billie Eilish, Labrinth, Ed Sheeran, Ruel, Lukas Graham, James Arthur, Lewis Capaldi, Lorcan Alred, and Elvis Presley. To the right, an Excel sheet displays the generated SQL scripts. The first few rows show the insertion of data into the 'singers' table:

```

1 • insert into singers Values('1','Low G','male','1997', null);
2 • insert into singers Values('2','Billie Eilish','female','2001', null);
3 • insert into singers Values('3','Labrinth','male','1989', null);
4
5 • SELECT * FROM musicsmodel.singers
    
```

Below the table, the output window shows the execution of these SQL statements, resulting in 10 rows affected.

- **UPDATE:**

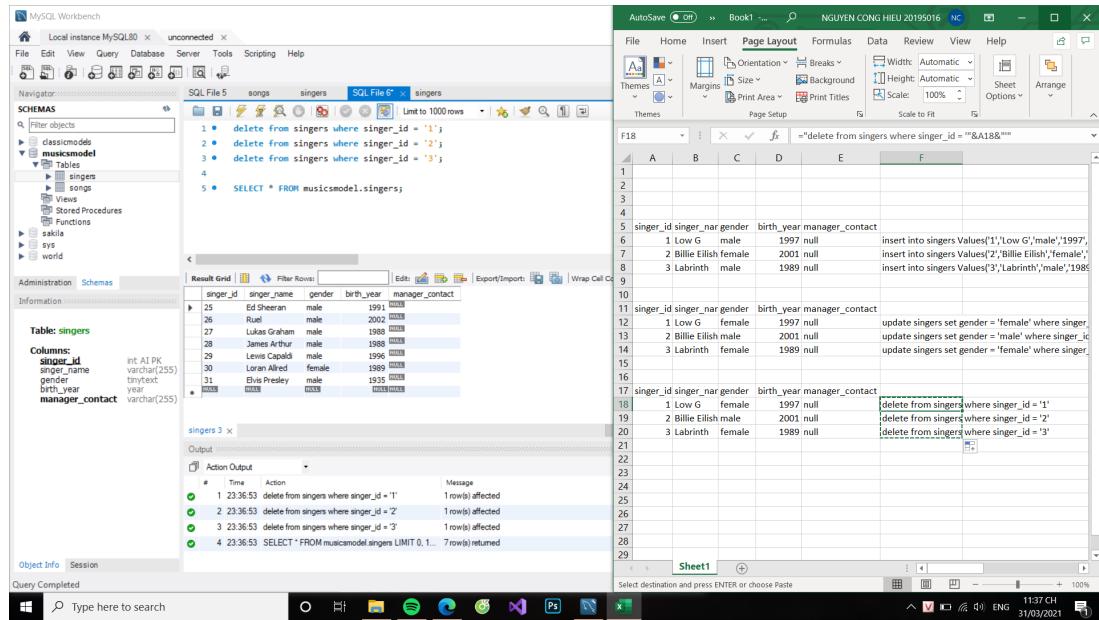
The screenshot shows the MySQL Workbench interface with the 'singers' table selected. The table structure and data are identical to the previous screenshot. The data grid shows the same 10 rows. To the right, an Excel sheet displays the generated SQL scripts for updating the 'gender' column. The first few rows show the update of the 'gender' column for specific rows:

```

1 • update singers set gender = 'female' where singer_id = '1';
2 • update singers set gender = 'male' where singer_id = '2';
3 • update singers set gender = 'female' where singer_id = '3';
4
5 • SELECT * FROM musicsmodel.singers;
    
```

Below the table, the output window shows the execution of these SQL statements, resulting in 1 row(s) affected for each update command.

- DELETE:



1.5.3 Bài 3

Cập nhật dữ liệu từ một bảng khác

INPUT: Ở bảng products tăng buyPrice lên 100 lần, với điều kiện là orderLineNumber = 1 và quantityInStock < 3000 trong bảng orderdetails.

OUTPUT: Bảng products đã được cập nhật.

SOURCE CODE:

```

USE classicmodels;

SET SQL_SAFE_UPDATES = 0;

UPDATE products
INNER JOIN orderdetails ON products.productCode =
    orderdetails.productCode
SET
    buyPrice = buyPrice * 100
WHERE
    quantityInStock < 3000 AND orderLineNumber = 1;

SELECT * FROM products;
    
```

ALGORITHM: Sử dụng bảng products join với bảng orderdetails trên cột productCode. Đặt buyPrice=buyPrice·100 với điều kiện quantityInStock nhỏ hơn 3000 và orderLineNumber bằng 1.

DISPLAY RESULTS:

| productCode | productName | productLine | productScale | productVendor | productDescription | quantityInStock | buyPrice | MSRP |
|-------------|---------------------------------------|--------------|--------------|--------------------------|--|-----------------|----------|--------|
| S10_1678 | 1969 Harley Davidson Ultimate Chopper | Motorcycles | 1:10 | Min Lin Diecast | This replica features working kickstand, front su... | 7933 | 48.81 | 95.70 |
| S10_1949 | 1952 Alpine Renault 1300 | Classic Cars | 1:10 | Classic Metal Creations | Turnable front wheels; steering function; detail... | 7305 | 98.58 | 214.30 |
| S10_2016 | 1996 Moto Guzzi 1100i | Motorcycles | 1:10 | Highway 66 Mini Classics | Official Moto Guzzi logos and insignias; saddle b... | 6625 | 68.99 | 118.94 |
| S10_4698 | 2003 Harley-Davidson Eagle Drag Bike | Motorcycles | 1:10 | Red Start Diecast | Model features, official Harley Davidson logos a... | 5582 | 91.02 | 193.66 |
| S10_4757 | 1972 Alfa Romeo GTA | Classic Cars | 1:10 | Motor City Art Classics | Features include: Turnable front wheels; steeri... | 3252 | 85.68 | 136.00 |
| S10_4962 | 1962 Lancia Delta 16V | Classic Cars | 1:10 | Second Gear Diecast | Features include: Turnable front wheels; steeri... | 6791 | 103.42 | 147.74 |
| S12_1099 | 1968 Ford Mustang | Classic Cars | 1:12 | Autoart Studio Design | Hood, doors and trunk all open to reveal highly ... | 68 | 9534.00 | 194.57 |
| S12_1108 | 2001 Ferrari Enzo | Classic Cars | 1:12 | Second Gear Diecast | Turnable front wheels; steering function; detail... | 3619 | 95.50 | 207.80 |

1.5.4 Bài 4

Cập nhật dữ liệu qua một procedure

INPUT: Dùng musicsmodel. Tạo một procedure để insert dữ liệu vào bảng singers. Truyền các tham số để cập nhật bảng singers.

OUTPUT: Bảng singers đã được update.

SOURCE CODE:

```
USE musicsmodel;
DELIMITER &&
CREATE PROCEDURE insertData (IN singer_name varchar(255), IN gender
tinytext, IN birth_year year, IN email varchar(255))

BEGIN
    INSERT INTO singers (singer_name, gender, debute_year,
    manager_contact)
    VALUES (singer_name, gender, birth_year, email);
END &&

DELIMITER ;

CALL insertData ('HieuNguyen', 'female', '2001',
'hieu.nc195016@sis.hust.edu.vn');

SELECT * FROM singers;
```

ALGORITHM: Tạo một procedure tên insertData với các tham số singer_name, gender, birth_year, email với các kiểu dữ liệu tương ứng với các cột trong bảng singers. Mỗi lần muốn thêm một bản ghi thì dùng lệnh CALL để gọi đến procedure và truyền vào các tham số tương ứng.

DISPLAY RESULTS:

| singer_id | singer_name | gender | birth_year | manager_contact |
|-----------|----------------|--------|------------|-------------------------------|
| 25 | Ed Sheeran | male | 1991 | NULL |
| 26 | Ruel | male | 2002 | NULL |
| 27 | Lukas Graham | male | 1988 | NULL |
| 28 | James Arthur | male | 1988 | NULL |
| 29 | Leviss Capaldi | male | 1996 | NULL |
| 30 | Loran Allred | female | 1985 | NULL |
| 31 | Elvis Presley | male | 1935 | NULL |
| 33 | HieuNguyen | female | 2001 | hieu.nc195016@sis.hust.edu.vn |
| * | NULL | NULL | NULL | NULL |

singers 1 ×

1.5.5 Bài 5

Tự động sinh ra các procedure cập nhật dữ liệu

INPUT: Có thể tự tạo hoặc dùng các phần mềm thứ 3.

OUTPUT: Trả về các procedure.

ALGORITHM: link phần mềm thứ 3.

DISPLAY RESULTS:

The screenshot shows a Windows desktop environment. In the foreground, there is a code editor window titled 'Account_Procedures.sql - StoredProceduresMySQL.cs.t Output'. The code contains SQL scripts for creating stored procedures. The first procedure is 'usp_SelectAccount' which takes an integer parameter 'pAccountID' and returns account details. The second procedure is 'usp_SelectAccountsByUniqueID' which takes an integer parameter 'pUniqueID' and returns accounts by unique ID. Both procedures are authored by 'Phosphor' and created on '01 Tháng Tư 2021'. Below the code editor is a 'CodeSmith' configuration dialog box. It has sections for 'ProcedureName' (set to 'usp_'), 'OutputFile' (set to 'tbl_'), and '3. Procedure Types'. Under '3. Procedure Types', several checkboxes are checked: 'IncludeDelete', 'IncludeDeleteByKey', 'IncludeInsert', 'IncludeInsertUpdate', 'IncludeSelect', 'IncludeSelectAll', 'IncludeSelectByKey', and 'IncludeUpdate'. At the bottom of the dialog is a 'Generate' button. The taskbar at the bottom of the screen shows various application icons, and the system tray indicates the date as '01/04/2021' and time as '12:59 SA'.

```
--region `usp_SelectAccount`

-- Author: Phosphor
-- Procedure Name: `usp_SelectAccount`
-- Date: 01 Tháng Tư 2021

CREATE PROCEDURE `usp_SelectAccount`
    (IN `pAccountID` Int)
SELECT
    `AccountID`,
    `UniqueId`,
    `Email`,
    `FirstName`,
    `LastName`,
    `Address1`,
    `Address2`,
    `City`,
    `State`,
    `Zip`,
    `Country`,
    `Phone`
FROM
    `Account`
WHERE
    `AccountID` = pAccountID;
--endregion

--region `usp_SelectAccountsByUniqueID`

-- Author: Phosphor
-- Procedure Name: `usp_SelectAccountsByUniqueID`
-- Date: 01 Tháng Tư 2021

CREATE PROCEDURE `usp_SelectAccountsByUniqueID`
    (IN `pUniqueID` Int)
```

1.6 Tuần 6

1.6.1 Bài 1

Aggregate Functions, Math Functions, Comparison Functions

INPUT:

- Aggregate Functions:
 - Hàm AVG: Tính giá trung bình của từng dòng sản phẩm.
 - Hàm SUM: Tính tổng giá trị số lượng các sản phẩm đã được đặt hàng.
 - Hàm MAX: Tìm giá sản phẩm cao nhất của từng dòng sản phẩm
- Math Function: Kiểm tra số lượng đặt hàng của từng sản phẩm là chẵn hay lẻ.
- Comparison Function: Thay thế các giá trị NULL ở phần comments trong bảng order thành 'There are no comments'.

OUTPUT:

- Aggregate Functions:
 - Hàm AVG: Bảng gồm 3 cột productLine, amount và average_Price.
 - Hàm SUM: Bảng gồm 3 cột productCode, productName và total.
 - Hàm MAX: Bảng gồm 2 cột productLine và MAX.
- Math Function: Bảng gồm 3 cột productName, total và OddOrEven.
- Comparison Function: Bảng gồm 2 cột orderNumber và comments.

SOURCE CODE:

- Aggregate Functions:

- Hàm AVG:

```

SELECT
    productLine,
    COUNT(*) AS amount,
    AVG(buyPrice) AS average_Price
FROM
    products
GROUP BY
    productLine;

```

- Hàm SUM:

```
SELECT
    productCode,
    productName,
    SUM(priceEach * quantityOrdered) AS total
FROM
    orderdetails INNER JOIN products USING (productCode)
GROUP BY productCode
ORDER BY total;
```

- Hàm MAX:

```
SELECT
    productLine,
    MAX(buyPrice)
FROM products
GROUP BY productLine
ORDER BY productLine;
```

- Math Functions:

```
SELECT
    productName,
    SUM(quantityOrdered) AS total,
    IF(MOD(SUM(quantityOrdered), 2),
        'Odd',
        'Even') OddOrEven
FROM
    orderdetails INNER JOIN products USING (productCode)
GROUP BY
    productCode
ORDER BY
    productCode;
```

- Comparison Functions:

```
SELECT
    orderNumber,
    COALESCE(comments, 'There are no comments') comments
FROM
    orders
ORDER BY orderNumber;
```

ALGORITHM:

- Aggregate Functions:

- Hàm AVG: Từ bảng products, nhóm theo productLine và lấy ra các thông tin như productLine, số lượng các dòng xe, và giá trung bình các dòng xe.
- Hàm SUM: Từ bảng orderdetails join với bảng products trên productCode và nhóm theo productCode. Lấy ra các thông tin như productCode, productName, và tổng tiền các sản phẩm được đặt hàng.
- Hàm MAX: Từ bảng products, nhóm theo productLine, dùng hàm max để tìm sản phẩm có giá cao nhất của mỗi dòng sản phẩm và sắp xếp theo productLine.
- Math Function: Từ bảng orderdetails join với bảng products sử dụng productCode, nhóm theo productCode. Lấy ra các thông tin về productName, tổng số lượng hàng đặt của mỗi sản phẩm, và xem tổng số lượng đó là chẵn hay lẻ('Even' - 'Odd').
- Comparison Function: Từ bảng orders lấy ra orderNumber. Kiểm tra xem nếu comments là NULL thì sẽ thay thế bằng 'There are no comments'.

DISPLAY RESULTS:

| productLine | amount | average_Price |
|------------------|--------|---------------|
| Classic Cars | 38 | 64.446316 |
| Motorcycles | 13 | 50.685385 |
| Planes | 12 | 49.629167 |
| Ships | 9 | 47.007778 |
| Trains | 3 | 43.923333 |
| Trucks and Buses | 11 | 56.329091 |
| Vintage Cars | 24 | 46.066250 |

Result 4 ×

| productCode | productName | total |
|-------------|---|----------|
| S24_1937 | 1939 Chevrolet Deluxe Coupe | 28052.94 |
| S24_3969 | 1936 Mercedes Benz 500k Roadster | 29763.39 |
| S24_2972 | 1982 Lamborghini Diablo | 30972.87 |
| S24_2840 | 1958 Chevy Corvette Limited Edition | 31627.96 |
| S32_2206 | 1982 Ducati 996 R | 33268.76 |
| S24_2022 | 1938 Cadillac V-16 Presidential Limousine | 38449.09 |
| S50_1341 | 1930 Buick Marquette Phaeton | 41599.24 |
| S24_1628 | 1966 Shelby Cobra 427 S/C | 42015.54 |
| S72_1253 | Boeing X-32A JSF | 42692.53 |
| S18_4668 | 1939 Cadillac Limousine | 44037.84 |

Result 2 ×

| productLine | MAX(buyPrice) |
|------------------|---------------|
| Classic Cars | 103.42 |
| Motorcycles | 91.02 |
| Planes | 77.27 |
| Ships | 82.34 |
| Trains | 67.56 |
| Trucks and Buses | 84.76 |
| Vintage Cars | 86.70 |

Result 3 ×

(a) Hàm AVG

(b) Hàm SUM

(c) Hàm MAX

Hình 1.5: Aggregate Functions

| productName | total | OddOrEven |
|---------------------------------------|-------|-----------|
| 1969 Harley Davidson Ultimate Chopper | 1057 | Odd |
| 1952 Alpine Renault 1300 | 961 | Odd |
| 1996 Moto Guzzi 1100i | 999 | Odd |
| 2003 Harley-Davidson Eagle Drag Bike | 985 | Odd |
| 1972 Alfa Romeo GTA | 1030 | Even |
| 1962 Lancia A Delta 16V | 932 | Even |
| 1968 Ford Mustang | 933 | Odd |
| 2004 Ferrari F430 | 1010 | Odd |

Result 5 ×

| orderNumber | comments |
|-------------|------------------------|
| 10100 | There are no comments |
| 10101 | Check on availability. |
| 10102 | There are no comments |
| 10103 | There are no comments |
| 10104 | There are no comments |
| 10105 | There are no comments |
| 10106 | There are no comments |
| 10107 | There are no comments |

Result 6 ×

(a) Hàm MOD

(b) Hàm COALESCE

Hình 1.6: Math & Comparison Function

1.6.2 Bài 2

Control Flow Functions and Expressions

INPUT: Thực hiện 2 truy vấn. Thứ nhất, tìm ra sức mua của từng khách hàng. Thứ hai, đếm những trạng thái(status) trong bảng orders.

OUTPUT:

- Truy vấn thứ nhất, là bảng gồm 3 cột customerName, orderCount và customerType.
- Truy vấn thứ hai, là bảng gồm 6 cột cancelled, disputed, inprogress, onhold, resolved, shipped.

SOURCE CODE:

```

WITH cte AS (
    SELECT
        customerName,
        COUNT(*) orderCount
    FROM
        orders INNER JOIN customers USING (customerNumber)
    GROUP BY customerName
)

SELECT
    customerName,
    orderCount,
    CASE orderCount
        WHEN 1 THEN 'One-time customer'
        WHEN 2 THEN 'Repeated customer'
        WHEN 3 THEN 'Frequent customer'
        ELSE 'Loyal customer'
    END AS customerType
FROM
    cte
ORDER BY customerName;

SELECT
    COUNT(IF(status = 'Cancelled', 1, NULL)) as cacelled,
    COUNT(IF(status = 'Disputed', 1, NULL)) as disputed,
    COUNT(IF(status = 'In Process', 1, NULL)) as inprogress,
    COUNT(IF(status = 'On Hold', 1, NULL)) as onhold,
    COUNT(IF(status = 'Resolved', 1, NULL)) as resolved,
    COUNT(IF(status = 'Shipped', 1, NULL)) as shipped
FROM
    orders;

```

ALGORITHM:

- Bước 1: Tạo một truy vấn con cte từ bảng orders join với customers trên customerNumber và nhóm theo customerName. Sau đó lấy ra các thông tin như customerName, orderCount(= số order của một khách hàng).
- Bước 2: Thực hiện truy vấn đầu tiên. Từ bảng cte lấy ra các cột customerName, orderCount, và customertype. Ở cột customerType sẽ nhận 4 giá trị 'One-time customer' nếu orderCount bằng 1, 'Repeated customer' nếu orderCount bằng 2, 'Frequent customer' nếu orderCount bằng 3, còn lại sẽ là 'Loyal customer'.
- Bước 3: Thực hiện truy vấn thứ 2. Từ bảng orders, đếm những giá trị giống nhau ở cột status và cho vào những cột tương ứng.

DISPLAY RESULTS:

| | customerName | orderCount | customerType |
|---|------------------------------|------------|-------------------|
| ▶ | Alpha Cognac | 3 | Frequent customer |
| | Amica Models & Co. | 2 | Repeated customer |
| | Anna's Decorations, Ltd | 4 | Loyal customer |
| | Atelier graphique | 3 | Frequent customer |
| | Australian Collectables, Ltd | 3 | Frequent customer |
| | Australian Collectors, Co. | 5 | Loyal customer |
| | Australian Gift Network, Co | 3 | Frequent customer |
| | Auto Associés & Cie. | 2 | Repeated customer |
| | Auto Glass Pros | 7 | Frequent customer |

Result 8 × Result 9

| | canceled | disputed | inprogress | onhold | resolved | shipped |
|---|----------|----------|------------|--------|----------|---------|
| ▶ | 6 | 3 | 12 | 4 | 4 | 303 |

(a) Truy vấn thứ nhất

(b) Truy vấn thứ hai

Hình 1.7: Các truy vấn

1.6.3 Bài 3

Window Functions

INPUT: Xếp bậc của khách hàng theo từng năm dựa trên số tiền thanh toán của khách hàng.

OUTPUT: Một bảng gồm các cột customerNumber, customerName, pay_year, amount, pay_rank.

SOURCE CODE:

```

SELECT
    customerNumber,
    customerName,
    YEAR(paymentDate) AS pay_year,
    amount,

```

```
DENSE_RANK() OVER (PARTITION BY YEAR(paymentDate) ORDER BY amount
DESC) AS pay_rank
FROM
payments INNER JOIN customers USING (customerNumber);
```

ALGORITHM: Từ bảng payments join với bảng customers trên customerNumber. Lấy các thông tin như customerNumber, customerName, pay_year, pay_rank. Trong đó, pay_year là năm thanh toán, pay_rank được xác định qua mức giảm dần của amount và phân vùng theo năm.

DISPLAY RESULTS:

| | customerNumber | customerName | pay_year | amount | pay_rank |
|-----|------------------------------|--------------|-----------|--------|----------|
| 323 | Down Under Souveniers, Inc | 2003 | 2880.00 | 97 | |
| 381 | Royale Belge | 2003 | 1627.56 | 98 | |
| 121 | Baane Mini Imports | 2003 | 1491.38 | 99 | |
| 381 | Royale Belge | 2003 | 1128.20 | 100 | |
| 141 | Euro + Shopping Channel | 2004 | 116208.40 | 1 | |
| 124 | Mini Gifts Distributors Ltd. | 2004 | 85410.87 | 2 | |
| 114 | Australian Collectors, Co. | 2004 | 82261.22 | 3 | |
| 239 | Collectable Mini Designs Co. | 2004 | 80375.24 | 4 | |
| 167 | Discount Classics Ltd. | 2004 | 80375.24 | 5 | |

1.6.4 Bài 4

Date Functions, String Functions

INPUT:

- Date Functions: Thực hiện 2 truy vấn. Thứ nhất, tìm ra số ngày để sản phẩm được giao đến khách hàng. Thứ hai, tìm ra số lượt order trong các ngày trong tuần vào năm 2004.
- String Function: Tổng số tiền thu được khi bán một sản phẩm(stock_value).

OUTPUT:

- Date Functions:

- Hàm DATEDIFF: Một bảng gồm 3 cột orderNumber, customerName, time_cost.
- Hàm DAYNAME: Một bảng gồm 2 cột weekday, total_order.

- String Functions: Một bảng gồm 2 cột productName, stock_value.

SOURCE CODE:

- Date Functions:

```

SELECT
    orderNumber,
    customerName,
    DATEDIFF(requiredDate, shippedDate) AS time_cost
FROM
    orders INNER JOIN customers USING (customerNumber);

SELECT
    DAYNAME(orderDate) AS weekday,
    COUNT(*) AS total_order
FROM
    orders
WHERE
    YEAR(orderDate) = 2004
GROUP BY weekday
ORDER BY weekday;

```

- String Functions;

```

SELECT
    productName,
    CONCAT('$' , FORMAT(SUM(quantityOrdered * priceEach), 2))
        AS stock_value
FROM
    orderdetails INNER JOIN products USING (productCode)
GROUP BY productName
ORDER BY productName;

```

ALGORITHM:

- Date Functions:
 - Hàm DATEDIFF: Từ bảng orders join với bảng customers trên customerNumber, lấy ra orderNumber, customerName và time_cost. Trong đó, time_cost số ngày từ requireDate đến shippeDate.
 - Hàm DAYNAME: Từ bảng orders, lấy ra các ngày trong orderDate và dùng hàm count để đếm xem có bao nhiêu lượt order trong mỗi ngày(dùng group by) với điều kiện năm đang xét là 2014.
- String Function: Từ bảng orderdetails join với products trên productCode, nhóm theo productName, lấy ra các cột như productName, stock_value. Trong đó stock_value được xác định bởi ký hiệu \$ và tổng của số tiền phải thanh toán(quantityOrdered*priceEach) làm tròn đến 2 chữ số sau dấu phẩy và sắp xếp theo productName.

DISPLAY RESULTS:

| | orderNumber | customerName | time_cost |
|---|-------------|------------------------------|-----------|
| ▶ | 10100 | Online Diecast Creations Co. | 3 |
| | 10101 | Blauer See Auto, Co. | 7 |
| | 10102 | Vitachrome Inc. | 4 |
| | 10103 | Baane Mini Imports | 5 |
| | 10104 | Euro+ Shopping Channel | 8 |
| | 10105 | Danish Wholesale Imports | 9 |
| | 10106 | Rovelli Gifts | 3 |
| | 10107 | Land of Toys Inc. | 5 |
| | 10108 | Customer 8 | 4 |

Result 5 × Result 6

(a) Hàm DATEDIFF

| | weekday | total_order |
|---|-----------|-------------|
| ▶ | Friday | 35 |
| | Monday | 24 |
| | Saturday | 11 |
| | Sunday | 2 |
| | Thursday | 26 |
| | Tuesday | 24 |
| | Wednesday | 29 |

Result 5 Result 6 ×

(b) Hàm DAYNAME

Hình 1.8: Date Functions

| | productName | stock_value |
|---|-------------------------------------|--------------|
| ▶ | 18th century schooner | \$112,427.12 |
| | 18th Century Vintage Horse Carriage | \$85,328.57 |
| | 1900s Vintage Bi-Plane | \$58,434.07 |
| | 1900s Vintage Tri-Plane | \$68,276.35 |
| | 1903 Ford Model A | \$111,528.82 |
| | 1904 Buick Runabout | \$77,800.54 |
| | 1911 Ford Town Car | \$61,833.77 |
| | 1912 Ford Model T Delivery Wagon | \$77,239.92 |
| | 1912 Ford Model T Coupe | \$67,102.00 |

Result 7 ×

Hình 1.9: String Functions

1.6.5 Bài 5 và 6

Danh chỉ mục dữ liệu

INPUT: Thực hiện truy vấn trên cùng một query để so sánh thời gian trước khi indexing và sau khi indexing.

OUTPUT: Bảng đã được đánh chỉ mục

SOURCE CODE:

- Before indexing

```
EXPLAIN SELECT * FROM customers
WHERE customerName = 'Corrida Auto Replicas, Ltd';
```

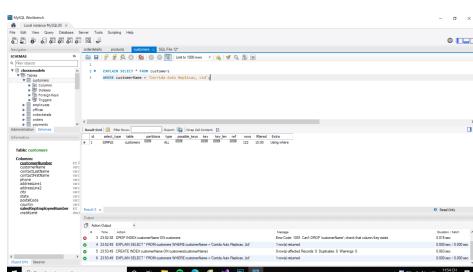
- After indexing

```
CREATE INDEX customerName ON customers(customerName);
```

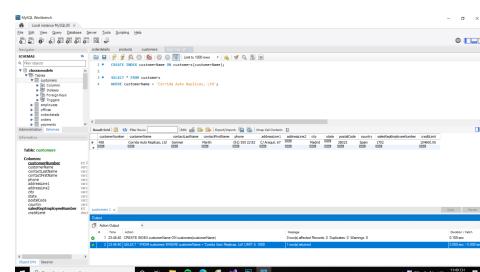
```
EXPLAIN SELECT * FROM customers
WHERE customerName = 'Corrida Auto Replicas, Ltd';
```

ALGORITHM:

DISPLAY RESULTS:



(a) Trước khi indexing



(b) Sau khi indexing

Hình 1.10: INDEXING

1.7 Tuần 7

1.7.1 Bài 1,2,3,4

Thiết kế CSDL lưu trữ đơn hàng

INPUT: Xây dựng lược đồ ER, RE.

OUTPUT: Tạo được CSDL lưu trữ đơn hàng.

SOURCE CODE:

```

SET AUTOCOMMIT = OFF;
START TRANSACTION;
CREATE DATABASE IF NOT EXISTS week7;
USE week7;

CREATE TABLE nguoi_dat_hang
(
    MaKhachHang INT AUTO_INCREMENT PRIMARY KEY,
    Ten VARCHAR(40) NOT NULL,
    SDT VARCHAR(15) NOT NULL
);

CREATE TABLE don_dat_hang
(
    MaDonHang INT AUTO_INCREMENT PRIMARY KEY,
    MaKhachHang INT,
    NgayDatHang DATE NOT NULL,
    DiaChiGiaoHang VARCHAR(255) NOT NULL
);

CREATE TABLE chi_tiet
(
    MaDonHang INT,
    ID_sp INT,
    SoLuong INT NOT NULL,
    PRIMARY KEY (MaDonHang, ID_sp)
);

CREATE TABLE mat_hang
(
    ID INT AUTO_INCREMENT PRIMARY KEY,
    TenHang VARCHAR(100) NOT NULL,
    Mota VARCHAR(255),
    DonVi VARCHAR(20) NOT NULL,
    Gia DECIMAL(10, 2) NOT NULL
);

```

```

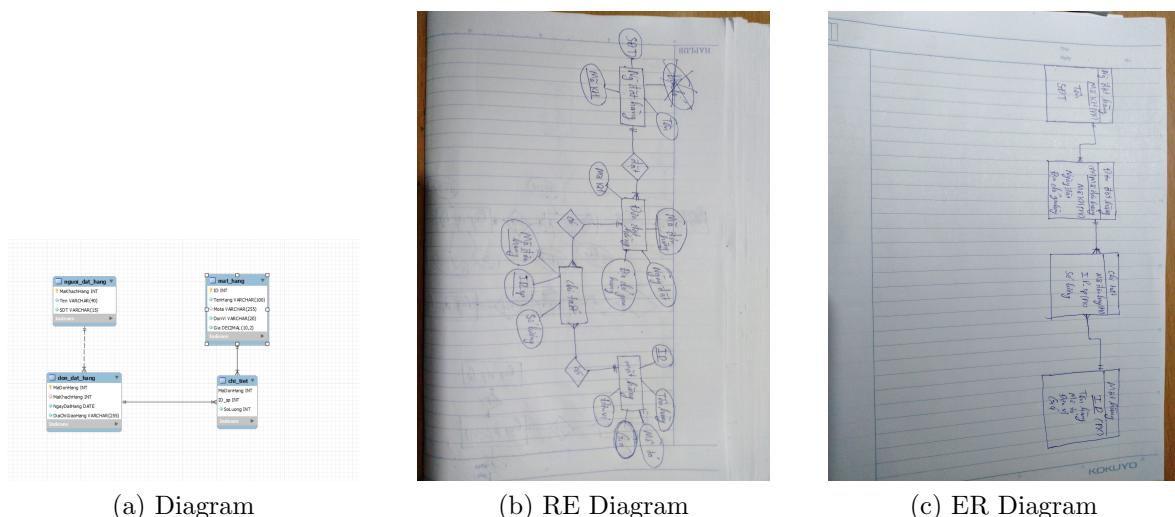
ALTER TABLE don_dat_hang
ADD CONSTRAINT MA_KH FOREIGN KEY (MaKhachHang) REFERENCES
nguoi_dat_hang (MaKhachHang);

ALTER TABLE chi_tiet
ADD CONSTRAINT fkof1 FOREIGN KEY (MaDonHang) REFERENCES don_dat_hang
(MaDonHang),
ADD CONSTRAINT fkof2 FOREIGN KEY (ID_sp) REFERENCES mat_hang (ID);

COMMIT;

```

ALGORITHM: DISPLAY RESULTS:



Hình 1.11: Diagram

1.7.2 Bài 5

Viết câu lệnh thêm dữ liệu vào bảng

INPUT: OUTPUT: Bảng đã được thêm dữ liệu **SOURCE CODE:**

```

USE week7;
SET AUTOCOMMIT = OFF;
START TRANSACTION;

INSERT INTO nguoi_dat_hang
VALUES
(1, 'Nguyen Van An', 987654321),
(2, 'Nguyen Thi B', 123456789),
(3, 'Hoang Linh Chi', 982158434);

```

```

INSERT INTO don_dat_hang
VALUES
(123, 1, '2009-11-18', '111 Nguyen Trai, Thanh Xuan, Ha Noi'),
(456, 2, '2009-10-05', 'khong biet'),
(789, 3, '2009-09-17', 'khong co');

INSERT INTO mat_hang
VALUES
(1, 'May tinh T450', 'May nhap moi', 'Chiec', 1000),
(2, 'Dien thoai Nokia5670', 'Dien thoai dang hot', 'Chiec', 200),
(3, 'May in Samsung450', 'May in dang e', 'Chiec', 100);

INSERT INTO chi_tiet
VALUES
(123, 1, 1),
(123, 2, 2),
(123, 3, 1),
(456, 1, 50),
(456, 2, 60),
(789, 1, 10),
(789, 2, 5),
(789, 3, 150);

COMMIT;

```

ALGORITHM: DISPLAY RESULTS:

| | MaKhachHang | Ten | SDT |
|---|-------------|----------------|-----------|
| ▶ | 1 | Nguyen Van An | 987654321 |
| | 2 | Nguyen Thi B | 123456789 |
| | 3 | Hoang Linh Chi | 982158434 |
| * | NULL | NULL | NULL |

1.7.3 Bài 6

Viết câu lệnh truy vấn

INPUT:

4. Viết các câu lệnh truy vấn để(Asignment 02):

1. Liệt kê danh sách khách hàng đã mua hàng ở cửa hàng.
2. Liệt kê danh sách sản phẩm của cửa hàng
3. Liệt kê danh sách các đơn đặt hàng của cửa hàng.

5. Viết các câu lệnh truy vấn để(Asignment 03):

1. Liệt kê danh sách khách hàng theo thứ tự alphabet.
2. Liệt kê danh sách khách hàng theo thứ tự ngược với alphabet.
3. Liệt kê danh sách sản phẩm của cửa hàng theo thứ tự giá giảm dần.
4. Liệt kê các sản phẩm của đơn hàng có mã là 123.

5. Liệt kê các sản phẩm mà khách hàng Nguyễn Văn An đã mua.

6. Viết các câu lệnh truy vấn để (Asignment 04):

1. Số khách hàng đã mua ở cửa hàng.
2. Số mặt hàng mà cửa hàng bán.
3. Tổng tiền của đơn hàng 123.
4. Tổng tiền của từng đơn hàng.
5. Hiển thị toàn bộ thông tin về khách hàng, của từng đơn hàng.

OUTPUT:**SOURCE CODE:**

- Truy vấn 1

```
SELECT TenHang
FROM mat_hang
INNER JOIN chi_tiet ON mat_hang.ID = chi_tiet.ID_sp
WHERE chi_tiet.MaDonHang = 123;
```

- Truy vấn 2

```
SELECT MaDonHang, TenHang
FROM chi_tiet
INNER JOIN mat_hang ON chi_tiet.ID_sp = mat_hang.ID;
```

- Truy vấn 3

```
SELECT * FROM don_dat_hang;
```

- Truy vấn 4

```
SELECT TenHang
FROM mat_hang;
```

- Truy vấn 5

```
SELECT * FROM nguoi_dat_hang
INNER JOIN don_dat_hang ON nguoi_dat_hang.MaKhachHang =
don_dat_hang.MaKhachHang;
```

- Truy vấn 6

```
SELECT tenhang
FROM mat_hang
ORDER BY tenhang DESC;
```

- Truy vấn 7

```
SELECT c1.Ten, c4.TenHang
FROM nguoi_dat_hang AS c1
INNER JOIN don_dat_hang AS c2
INNER JOIN chi_tiet AS c3
INNER JOIN mat_hang AS c4
ON c1.MaKhachHang = c2.MaKhachHang AND c2.MaDonHang =
c3.MaDonHang AND c3.ID_sp = c4.ID
WHERE Ten = 'Nguyen Van An';
```

- Truy vấn 8

```
SELECT COUNT(*) AS SoKhachHang
FROM nguoi_dat_hang;
```

- Truy vấn 9

```
SELECT COUNT(*)
FROM mat_hang;
```

- Truy vấn 10

```
SELECT MaDonHang, SUM(c2.Gia * c1.SoLuong) AS ThanhTien
FROM chi_tiet AS c1
INNER JOIN mat_hang AS c2 ON c1.ID_sp = c2.ID
WHERE c1.MaDonHang = 123;
```

- Truy vấn 11

```
SELECT MaDonHang, SUM(Gia*SoLuong) AS ThanhTien
FROM chi_tiet AS c1
INNER JOIN mat_hang AS c2 ON c1.ID_sp = c2.ID
```

GROUP BY c1.MaDonHang;

- Truy vấn 12

```
SELECT *
FROM don_dat_hang AS c1
INNER JOIN nguoi_dat_hang AS c2 USING (MaKhachHang);
```

ALGORITHM:

DISPLAY RESULTS:

- Truy vấn 1

| | TenHang |
|---|----------------------|
| ▶ | May tinh T450 |
| | Dien thoai Nokia5670 |
| | May in Samsung450 |

- Truy vấn 2

| | MaDonHang | TenHang |
|---|-----------|----------------------|
| ▶ | 123 | May tinh T450 |
| | 456 | May tinh T450 |
| | 789 | May tinh T450 |
| | 123 | Dien thoai Nokia5670 |
| | 456 | Dien thoai Nokia5670 |
| | 789 | Dien thoai Nokia5670 |
| | 123 | May in Samsung450 |
| | 789 | May in Samsung450 |

- Truy vấn 3

| | MaDonHang | MaKhachHang | NgayDatHang | DiaChiGiaoHang |
|---|-----------|-------------|-------------|-------------------------------------|
| ▶ | 123 | 1 | 2009-11-18 | 111 Nguyen Trai, Thanh Xuan, Ha Noi |
| | 456 | 2 | 2009-10-05 | khong biet |
| | 789 | 3 | 2009-09-17 | khong co |
| * | NULL | NULL | NULL | NULL |

- Truy vấn 4

| | TenHang |
|---|----------------------|
| ▶ | May tinh T450 |
| | Dien thoai Nokia5670 |
| | May in Samsung450 |

- Truy vấn 5

| | MaKhachHang | Ten | SDT | MaDonHang | MaKhachHang | NgayDatHang | DiaChiGiaoHang |
|---|-------------|----------------|-----------|-----------|-------------|-------------|-------------------------------------|
| ▶ | 1 | Nguyen Van An | 987654321 | 123 | 1 | 2009-11-18 | 111 Nguyen Trai, Thanh Xuan, Ha Noi |
| | 2 | Nguyen Thi B | 123456789 | 456 | 2 | 2009-10-05 | khong biet |
| | 3 | Hoang Linh Chi | 982158434 | 789 | 3 | 2009-09-17 | khong co |

- Truy vấn 6

| | |
|---|----------------------|
| | tenhang |
| ▶ | May tinh T450 |
| | May in Samsung450 |
| | Dien thoai Nokia5670 |

- Truy vấn 7

| | Ten | TenHang |
|---|---------------|----------------------|
| ▶ | Nguyen Van An | May tinh T450 |
| | Nguyen Van An | Dien thoai Nokia5670 |
| | Nguyen Van An | May in Samsung450 |

- Truy vấn 8

| | |
|---|-------------|
| | SokhachHang |
| ▶ | 3 |

- Truy vấn 9

| | |
|---|----------|
| | COUNT(*) |
| ▶ | 3 |

- Truy vấn 10

| | MaDonHang | ThanhTien |
|---|-----------|-----------|
| ▶ | 123 | 1500.00 |

- Truy vấn 11

| | MaDonHang | ThanhTien |
|---|-----------|-----------|
| ▶ | 123 | 1500.00 |
| | 456 | 62000.00 |
| | 789 | 26000.00 |

- Truy vấn 12

| | MaKhachHang | MaDonHang | NgayDatHang | DiaChiGiaoHang | Ten | SDT |
|---|-------------|-----------|-------------|-------------------------------------|----------------|-----------|
| ▶ | 1 | 123 | 2009-11-18 | 111 Nguyen Trai, Thanh Xuan, Ha Noi | Nguyen Van An | 987654321 |
| | 2 | 456 | 2009-10-05 | khong biet | Nguyen Thi B | 123456789 |
| | 3 | 789 | 2009-09-17 | khong co | Hoang Linh Chi | 982158434 |

1.7.4 Bài 7

Thay đổi cấu trúc CSDL

INPUT:

1. Viết câu lệnh để thay đổi trường điện thoại của khách hàng là *not null*.
2. Viết câu lệnh để thay đổi trường giá tiền của từng mặt hàng là dương(>0).
3. Viết câu lệnh để thay đổi ngày đặt hàng của khách hàng phải nhỏ hơn ngày hiện tại.

OUTPUT:

SOURCE CODE:

- 1 và 2:

```
USE week7;
ALTER TABLE nguoi_dat_hang
MODIFY SDT VARCHAR(30) NOT NULL;

ALTER TABLE mat_hang
MODIFY Gia INT UNSIGNED;
```

- 3:

```
DELIMITER $$

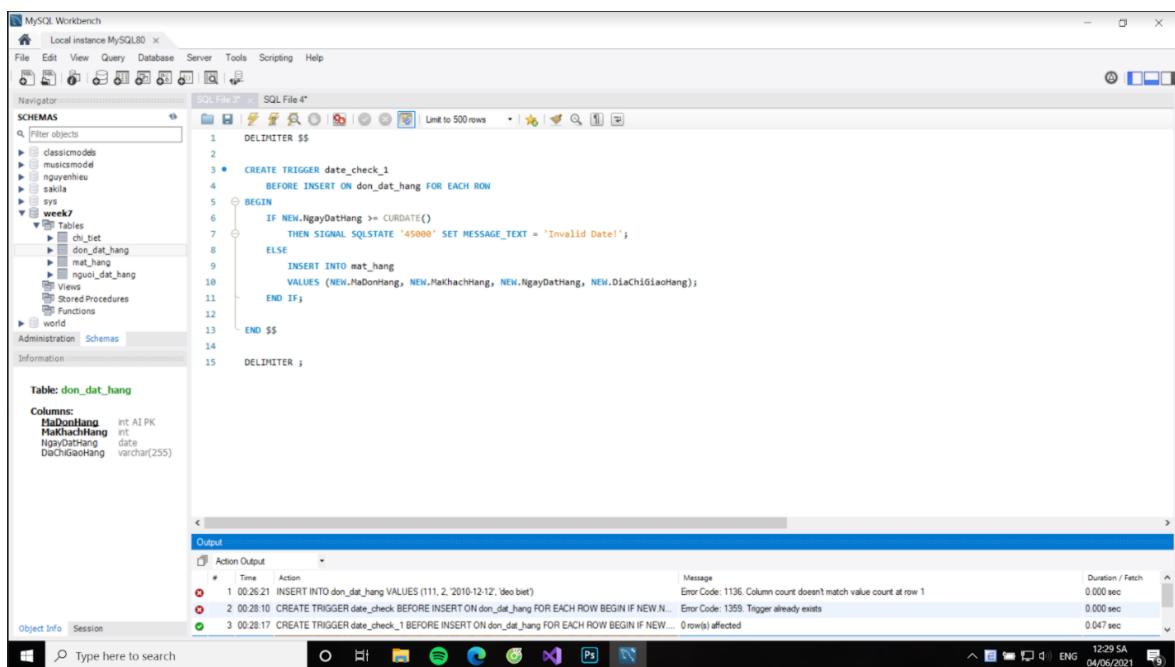
CREATE TRIGGER date_check_1
BEFORE INSERT ON don_dat_hang FOR EACH ROW
BEGIN
    IF NEW.NgayDatHang >= CURDATE()
        THEN SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Invalid
Date!';
    ELSE
        INSERT INTO mat_hang
        VALUES (NEW.MaDonHang, NEW.MaKhachHang, NEW.NgayDatHang,
        NEW.DiaChiGiaoHang);
    END IF;

END $$

DELIMITER ;
```

ALGORITHM: Ở câu 3, dùng trigger để trước khi insert dữ liệu vào bảng, ta sẽ kiểm tra xem ngày đặt của nó có nhỏ hơn ngày hiện tại không, nếu không thì in ra output là 'Invalid Date!', nếu thỏa mãn thì sẽ insert vào.

DISPLAY RESULTS:



1.8 Tuần 8

1.8.1 Bài 1

Bài 1. Cho quan hệ R trên tập thuộc tính $U = \{A, B, C, D, E, G\}$ và tập các phụ thuộc hàm $F = \{AB \rightarrow C, AC \rightarrow D, D \rightarrow EG, G \rightarrow B, A \rightarrow D, CG \rightarrow A\}$.

- Chứng minh rằng nếu R thoả mãn F thì R cũng thoả mãn các phụ thuộc hàm $AB \rightarrow E$ và $AD \rightarrow BC$.
- Tính bao đóng $\{A\}^+$ của tập thuộc tính $\{A\}$.
- Tập các phụ thuộc hàm F đã là tối thiểu chưa? Vì sao? Nếu chưa, tìm một phủ tối thiểu của F.

INPUT: $U = \{A, B, C, D, E, G\}$ và $F = \{AB \rightarrow C, AC \rightarrow D, D \rightarrow EG, G \rightarrow B, A \rightarrow D, CG \rightarrow A\}$

OUTPUT:

- a, $AB \rightarrow E$ và $AD \rightarrow BC$.
- b, $\{A\}^+$.
- c, Một phủ tối thiểu của F.

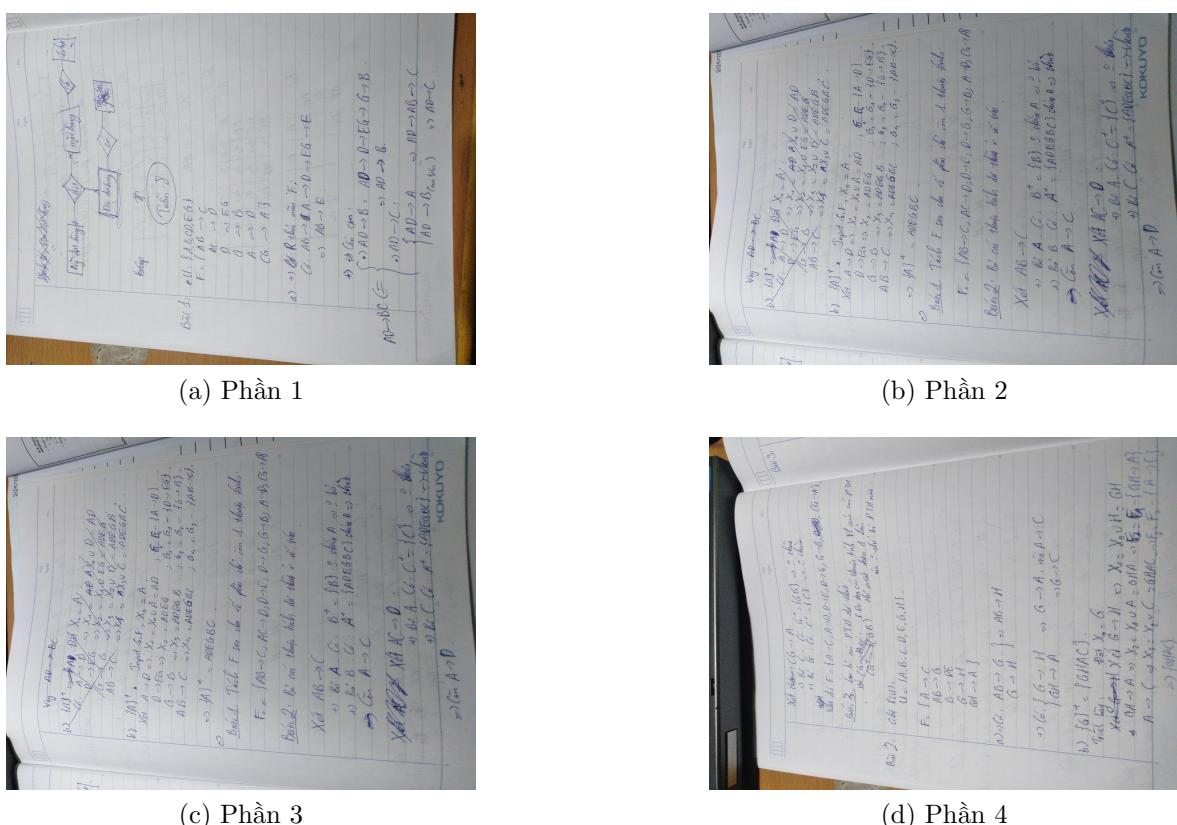
SOURCE CODE:

ALGORITHM:

- a, Dựa các các tiên đề Armstrong và hệ quả.

- b, Đặt $X_0 = A$ và $G = F$. Dựa vào các phụ thuộc hàm ta thêm vào các thuộc tính chưa có vào X_0 . Sau mỗi lần dùng một PTH ta loại nó ra khỏi G . Cuối cùng khi loại bỏ hết PTH hoặc không thể thêm được nữa ta có X_0 là bao đóng cần tìm.
- c, Gồm 3 bước. Bước 1: Tách F sao cho về phải chỉ còn 1 thuộc tính. Bước 2: Bỏ các thuộc tính dư thừa ở về trái. Bước 3: Loại bỏ các PTH dư thừa.

DISPLAY RESULTS:



Hình 1.12: Bài 1 (Link)

1.8.2 Bài 2

Bài 2. Cho quan hệ R(U): $U = \{A, B, C, D, E, G, H\}$ và tập phụ thuộc hàm $F = \{A \rightarrow C, AB \rightarrow G, B \rightarrow DE, G \rightarrow H, GH \rightarrow A\}$.

- Chứng minh rằng nếu R thoả mãn F thì R cũng thoả mãn các phụ thuộc hàm $AB \rightarrow H$ và $G \rightarrow C$.
- Tính bao đóng $\{G\}^+$ của tập thuộc tính $\{G\}$.

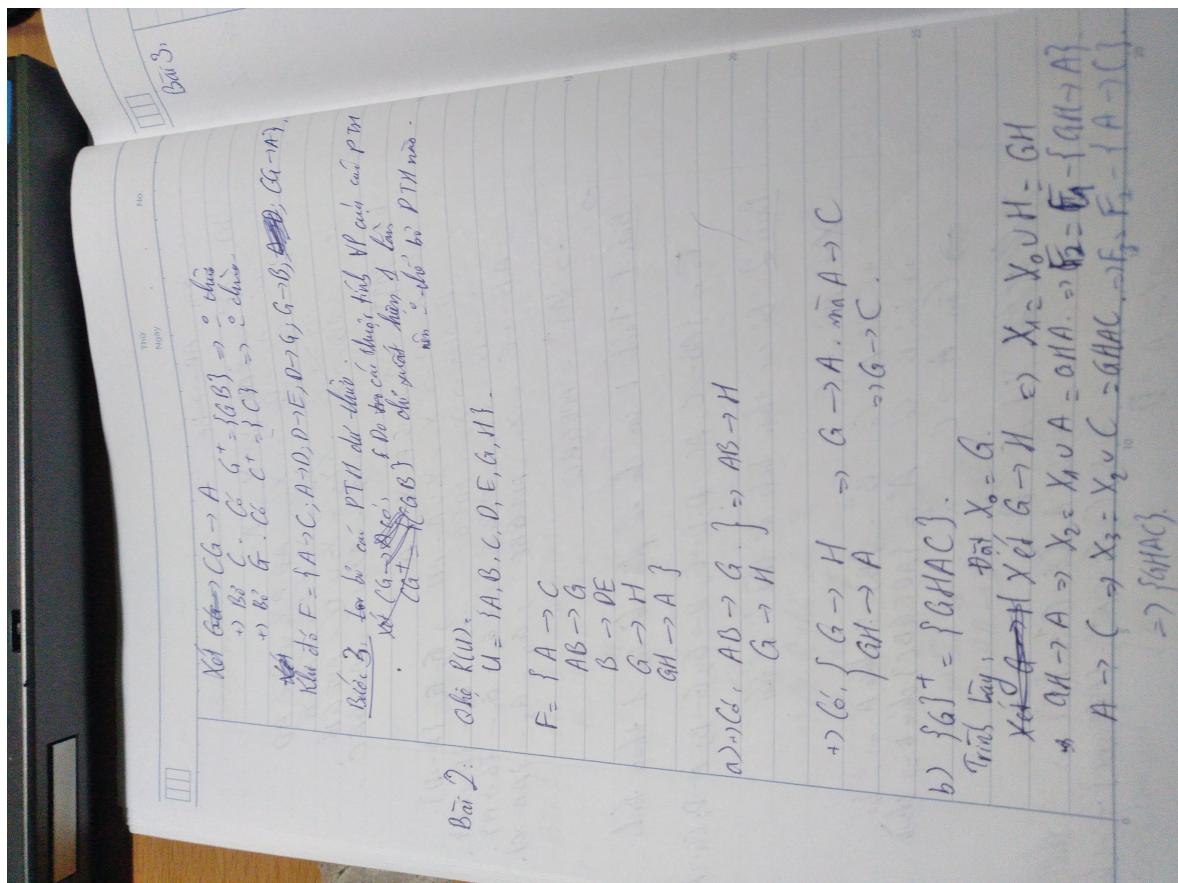
INPUT: $U = \{A, B, C, D, E, G, H\}$ và $F = \{A \rightarrow C, AB \rightarrow G, B \rightarrow DE, G \rightarrow H, GH \rightarrow A\}$

OUTPUT:

- a, $AB \rightarrow H$ và $G \rightarrow C$.
- b, $\{G\}^+$.

SOURCE CODE:**ALGORITHM:**

- a, Dựa các các tiên đề Armstrong và hệ quả.
- b, Đặt $X_0 = G$ và $I = F$. Dựa vào các phụ thuộc hàm ta thêm vào các thuộc tính chưa có vào X_0 . Sau mỗi lần dùng một PTH ta loại nó ra khỏi I. Cuối cùng khi loại bỏ hết PTH hoặc không thể thêm được nữa ta có X_0 là bao đóng cần tìm.

DISPLAY RESULTS:

(a) Phần 1

Hình 1.13: Bài 2 (Link)

1.8.3 Bài 3

Bài 3. Cho quan hệ R trên tập thuộc tính U = {G, H, I, K, L, M} và tập các phụ thuộc hàm F = {GH → L, I → M, L → K, HM → G, GK → I, H → L}.

- Chứng minh rằng nếu R thoả mãn F thì R cũng thoả mãn các phụ thuộc hàm HI → G và GH → KM.
- Tính bao đóng $\{G, H\}^+$ của tập thuộc tính {G, H}.
- Tập các phụ thuộc hàm F đã là tối thiểu chưa? Vì sao? Nếu chưa, tìm một phủ tối thiểu của F.

INPUT: U = {G, H, I, K, L, M} và F = {GH → L, I → M, L → K, HM → G, GK → I, H → L}

OUTPUT:

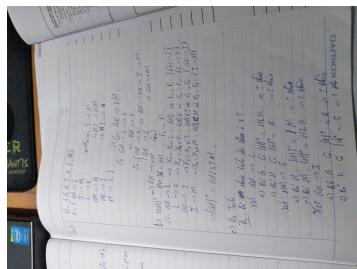
- a, HI → G và GH → KM.
- b, $\{GH\}^+$.
- c, Một phủ tối thiểu của F.

SOURCE CODE:

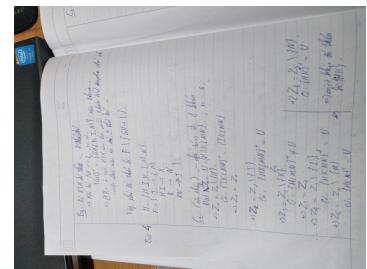
ALGORITHM:

- a, Dựa các các tiên đề Armstrong và hệ quả.
- b, Đặt $X_0 = GH$ và A = F. Dựa vào các phụ thuộc hàm ta thêm vào các thuộc tính chưa có vào X_0 . Sau mỗi lần dùng một PTH ta loại nó ra khỏi A. Cuối cùng khi loại bỏ hết PTH hoặc không thể thêm được nữa ta có X_0 là bao đóng cần tìm.
- c, Gồm 3 bước. Bước 1: Tách F sao cho về phải chỉ còn 1 thuộc tính. Bước 2: Bỏ các thuộc tính dư thừa ở về trái. Bước 3: Loại bỏ các PTH dư thừa.

DISPLAY RESULTS:



(a) Phần 1



(b) Phần 2

Hình 1.14: Bài 3 ([Link](#))

1.8.4 Bài 4

Bài 4. Cho quan hệ R(U): $U = \{H, I, K, L, M, N\}$ và tập phụ thuộc hàm $F = \{I \rightarrow LM, HI \rightarrow K, K \rightarrow N, KN \rightarrow I\}$.

- Tìm một khóa tối thiểu của quan hệ R.
- Kiểm tra tính mất mát thông tin khi tách R thành các quan hệ: $R_1(HIN)$; $R_2(HIK)$, $R_3(ILM)$.
- Chuẩn hóa quan hệ R về dạng chuẩn 3NF.

INPUT: $U = \{H, I, K, L, M, N\}$ và $F = \{I \rightarrow LM, HI \rightarrow K, K \rightarrow N, KN \rightarrow I\}$

OUTPUT:

- a, Một khóa tối thiểu của R.
- b, Kết luận tính mất mát thông tin của R sau khi tách.
- c, Chuẩn hóa về dạng 3NF.

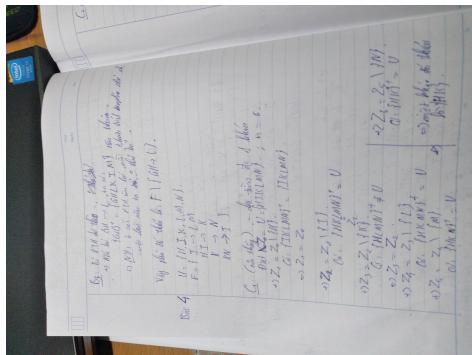
SOURCE CODE:

ALGORITHM:

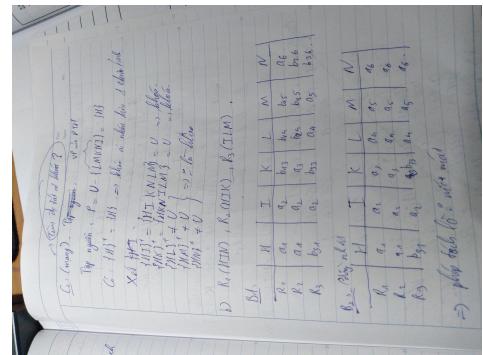
- a, Xác định tập nguồn $P = U - \{LMKNI\} = \{H\}$. Tìm $\{H\}^+ = \{H\} \Rightarrow$ khóa có nhiều hơn 1 thuộc tính. Xác định các $\{HI\}^+, \{HK\}^+, \{HL\}^+, \{HM\}^+, \{HN\}^+$. Nếu bằng U thì là khóa.
- b,
 - Bước 1: xây dựng bảng gồm 6 cột, 3 hàng với cột j tương ứng với thuộc tính U_j và hàng i tương ứng với sơ đồ R_i . Tại hàng i, cột j, nếu U_j thuộc R_i thi ta điền a_j ngược lại điền b_{ij} .
 - Bước 2: Xét lần lượt các PTH và áp dụng cho bảng vừa dựng. Giả sử xét PTH $X \longleftrightarrow Y$. Nếu tồn tại hai hàng mà tất cả các cột tương ứng với các thuộc tính của Y cũng có giá trị như nhau trong 2 hàng đó theo nguyên tắc:
 - * Nếu có 1 ký hiệu a_j trong các cột ứng với các thuộc tính Y thì đồng nhất các ký hiệu là a_j .
 - * Nếu không thì đồng nhất bằng 1 trong các ký hiệu b_{ij} .
 - Bước 3: Tiếp tục áp dụng các PTH cho bảng (kể cả việc lặp lại các PTH đã áp dụng) cho tới khi không thể thay đổi được giá trị nào trong bảng.
 - Bước 4: Nếu trong bảng có 1 hàng gồm các ký hiệu a_1, a_2, \dots, a_6 thì phép tách là không mất thông tin. Ngược lại thì phép tách không bảo toàn thông tin.

- c, Đây là bài toán đi tìm PTH đã xét ở Bài 3.

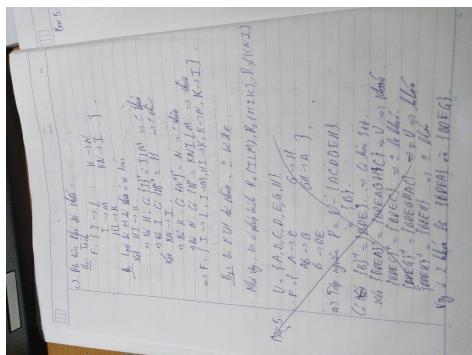
DISPLAY RESULTS:



(a) Phần 1



(b) Phần 2



(c) Phần 3

Hình 1.15: Bài 4 (Link)

1.8.5 Bài 5

Bài 5. Cho quan hệ R(U): U = { A, B, C, D, E, G, H } và tập phụ thuộc hàm F = { A → C, AB → G, B → DE, G → H, GH → A }.

- Tìm một khóa tối thiểu của quan hệ R.
- Kiểm tra tính mất mát thông tin khi tách R thành các quan hệ : R₁(AC); R₂(BDE); R₃(ABGH).
- Chuẩn hóa quan hệ R về dạng chuẩn 3NF.

INPUT: U = {A,B,C,D,E,F,G,H} và F = { A → C, AB → G, B → DE, G → H, GH → A }

OUTPUT:

- a, Một khóa tối thiểu của R.
- b, Kết luận tính mất mát thông tin của R sau khi tách.
- c, Chuẩn hóa về dạng 3NF.

SOURCE CODE:**ALGORITHM:**

- a, Xác định tập nguồn $P = U - \{ACDGH\} = \{B\}$. Tìm $\{B\}^+ = \{BDE\} \Rightarrow$ khóa có nhiều hơn 1 thuộc tính. Xác định các $\{BA\}^+, \{BG\}^+$. Nếu bằng U thì là khóa.
- b,
 - Bước 1: xây dựng bảng gồm 7 cột, 3 hàng với cột j tương ứng với thuộc tính U_j và hàng i tương ứng với sơ đồ R_i . Tại hàng i, cột j, nếu U_j thuộc R_i thi ta điền a_j ngược lại điền b_{ij} .
 - Bước 2: Xét lần lượt các PTH và áp dụng cho bảng vừa dựng. Giả sử xét PTH $X \longleftrightarrow Y$. Nếu tồn tại hai hàng mà tất cả các cột tương ứng với các thuộc tính của Y cũng có giá trị như nhau trong 2 hàng đó theo nguyên tắc:
 - * Nếu có 1 ký hiệu a_j trong các cột ứng với các thuộc tính Y thì đồng nhất các ký hiệu là a_j .
 - * Nếu không thì đồng nhất bằng 1 trong các ký hiệu b_{ij} .
 - Bước 3: Tiếp tục áp dụng các PTH cho bảng (kể cả việc lặp lại các PTH đã áp dụng) cho tới khi không thể thay đổi được giá trị nào trong bảng.
 - Bước 4: Nếu trong bảng có 1 hàng gồm các ký hiệu a_1, a_2, \dots, a_6 thì phép tách là không mất thông tin. Ngược lại thì phép tách không bảo toàn thông tin.
- c, Đây là bài toán đi tìm PTH đã xét ở Bài 3.

DISPLAY RESULTS:

| R_5 | $\bigvee_{\beta \in \{A, B, C, D, E, F, G, H\}} R_5 \beta \wedge \beta' \in \text{CAT}(f)$ |
|-------------------|--|
| $A \rightarrow B$ | $A \rightarrow B \wedge B' \in \text{CAT}(f)$ |
| $B \rightarrow C$ | $B \rightarrow C \wedge C' \in \text{CAT}(f)$ |
| $C \rightarrow D$ | $C \rightarrow D \wedge D' \in \text{CAT}(f)$ |
| $D \rightarrow E$ | $D \rightarrow E \wedge E' \in \text{CAT}(f)$ |
| $E \rightarrow F$ | $E \rightarrow F \wedge F' \in \text{CAT}(f)$ |
| $F \rightarrow G$ | $F \rightarrow G \wedge G' \in \text{CAT}(f)$ |
| $G \rightarrow H$ | $G \rightarrow H \wedge H' \in \text{CAT}(f)$ |

a) $\bigvee_{\beta \in \{A, B, C, D, E, F, G, H\}} R_5 \beta \wedge \beta' \in \text{CAT}(f)$ $\Rightarrow R_5 \beta \wedge \beta' \in \text{CAT}(f)$ $\forall \beta \in \{A, B, C, D, E, F, G, H\}$

b) $R_5(A \rightarrow B) \wedge R_5(B \rightarrow C) \wedge R_5(C \rightarrow D) \wedge R_5(D \rightarrow E) \wedge R_5(E \rightarrow F) \wedge R_5(F \rightarrow G) \wedge R_5(G \rightarrow H) \in \text{CAT}(f)$

(a) Phần 1

| R_5 | $\bigvee_{\beta \in \{A, B, C, D, E, F, G, H\}} R_5 \beta \wedge \beta' \in \text{CAT}(f)$ |
|-------------------|--|
| $A \rightarrow B$ | $A \rightarrow B \wedge B' \in \text{CAT}(f)$ |
| $B \rightarrow C$ | $B \rightarrow C \wedge C' \in \text{CAT}(f)$ |
| $C \rightarrow D$ | $C \rightarrow D \wedge D' \in \text{CAT}(f)$ |
| $D \rightarrow E$ | $D \rightarrow E \wedge E' \in \text{CAT}(f)$ |
| $E \rightarrow F$ | $E \rightarrow F \wedge F' \in \text{CAT}(f)$ |
| $F \rightarrow G$ | $F \rightarrow G \wedge G' \in \text{CAT}(f)$ |
| $G \rightarrow H$ | $G \rightarrow H \wedge H' \in \text{CAT}(f)$ |

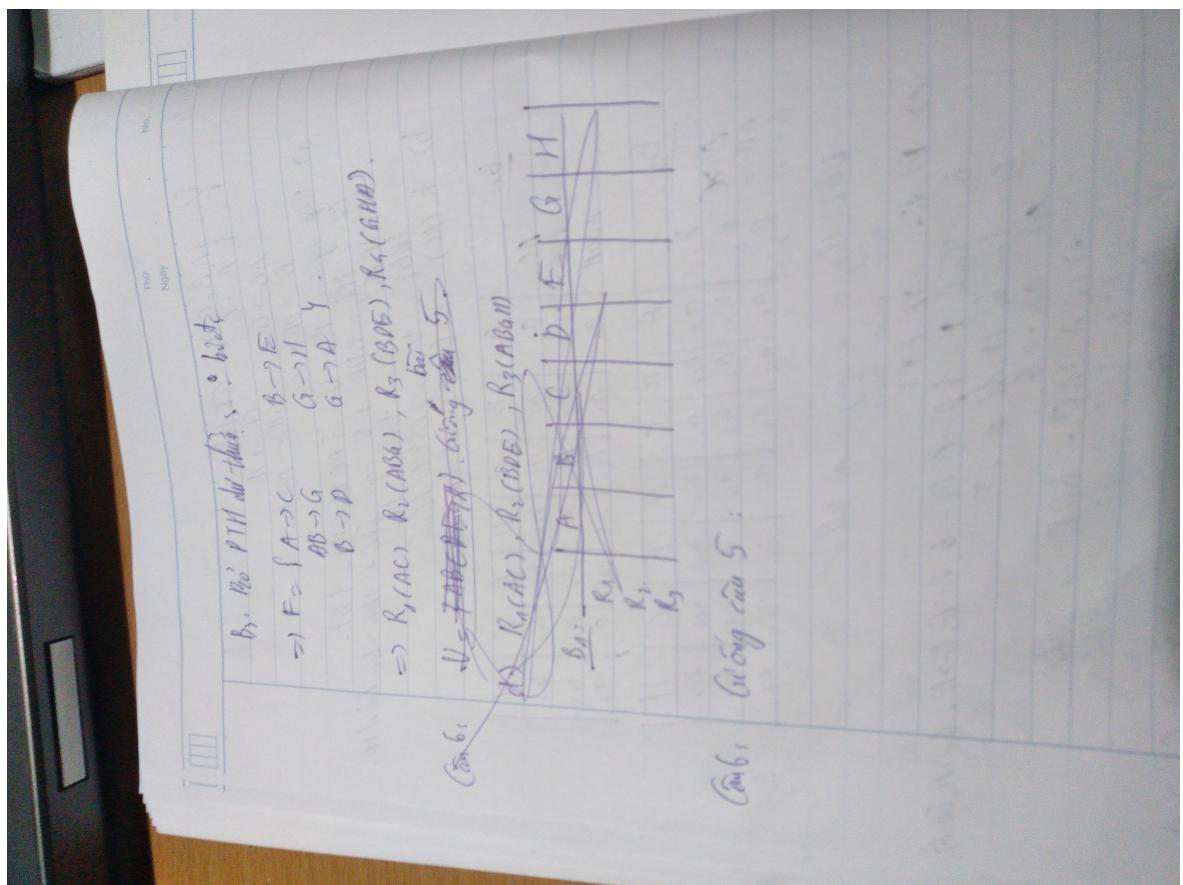
$\bigvee_{\beta \in \{A, B, C, D, E, F, G, H\}} R_5 \beta \wedge \beta' \in \text{CAT}(f)$ $\Rightarrow R_5(A \rightarrow B) \wedge R_5(B \rightarrow C) \wedge R_5(C \rightarrow D) \wedge R_5(D \rightarrow E) \wedge R_5(E \rightarrow F) \wedge R_5(F \rightarrow G) \wedge R_5(G \rightarrow H) \in \text{CAT}(f)$

(b) Phần 2

Hình 1.16: Bài 5 (Link)

1.8.6 Bài 6

Giống bài 5



1.9 Tuần 9

1.9.1 Bài 1

Start, Stop, and Restart MySQL Server

SOURCE CODE:

- Start: net START MYSQL80
- Stop: net STOP MYSQL80
- Restart:

DISPLAY RESULTS:



Hình 1.17: Bài 1

1.9.2 Bài 2

Users, Roles, and Privileges

SOURCE CODE:

- Create roles:

```
CREATE ROLE
    crm_dev,
    crm_read,
    crm_write;
```

- Create users:

```
CREATE USER bob@localhost identified by 'Secure1pass!';
```

- Change role password:

```
SET PASSWORD FOR 'dbadmin'@'localhost' = PASSWORD('bigshark');
```

- Drop user:

```
drop user dbadmin@localhost
```

- Grant and Assign roles:

```
GRANT INSERT  
ON classicmodels.*  
TO bob@localhost;
```

- Grant and Revoke privileges:

```
REVOKE INSERT, UPDATE  
ON classicmodels.*  
FROM rfc@localhost;
```

DISPLAY RESULTS:

```
Administrator: Command Prompt - MySQL - root@127.0.0.1 [classicmodels]
CREATE USER 'bob'@'localhost' IDENTIFIED BY 'bigshark';
Query OK, 0 rows affected (0.00 sec)

bob@bob's MySQL shell
```

(a) Create roles

```
Administrator: Command Prompt - MySQL - root@127.0.0.1 [classicmodels]
CREATE USER 'bob'@'localhost' IDENTIFIED BY 'bigshark';
Query OK, 0 rows affected (0.00 sec)

bob@bob's MySQL shell
```

(b) Create users

```
Administrator: Command Prompt - MySQL - root@127.0.0.1 [classicmodels]
SET PASSWORD FOR 'bob'@'localhost' = PASSWORD('bigshark');
Query OK, 0 rows affected (0.00 sec)

bob@bob's MySQL shell
```

(c) Change role password

```
Administrator: Command Prompt - MySQL - root@127.0.0.1 [classicmodels]
DROP USER bob@localhost;
Query OK, 0 rows affected (0.00 sec)

bob@bob's MySQL shell
```

(d) Drop user

```
Administrator: Command Prompt - MySQL - root@127.0.0.1 [classicmodels]
GRANT INSERT ON classicmodels.* TO bob@localhost;
Query OK, 0 rows affected (0.00 sec)

bob@bob's MySQL shell
```

(e) Grant and Assign roles

```
Administrator: Command Prompt - MySQL - root@127.0.0.1 [classicmodels]
REVOKE INSERT, UPDATE ON classicmodels.* FROM bob@localhost;
Query OK, 0 rows affected (0.00 sec)

bob@bob's MySQL shell
```

(f) Grant and Revoke privileges

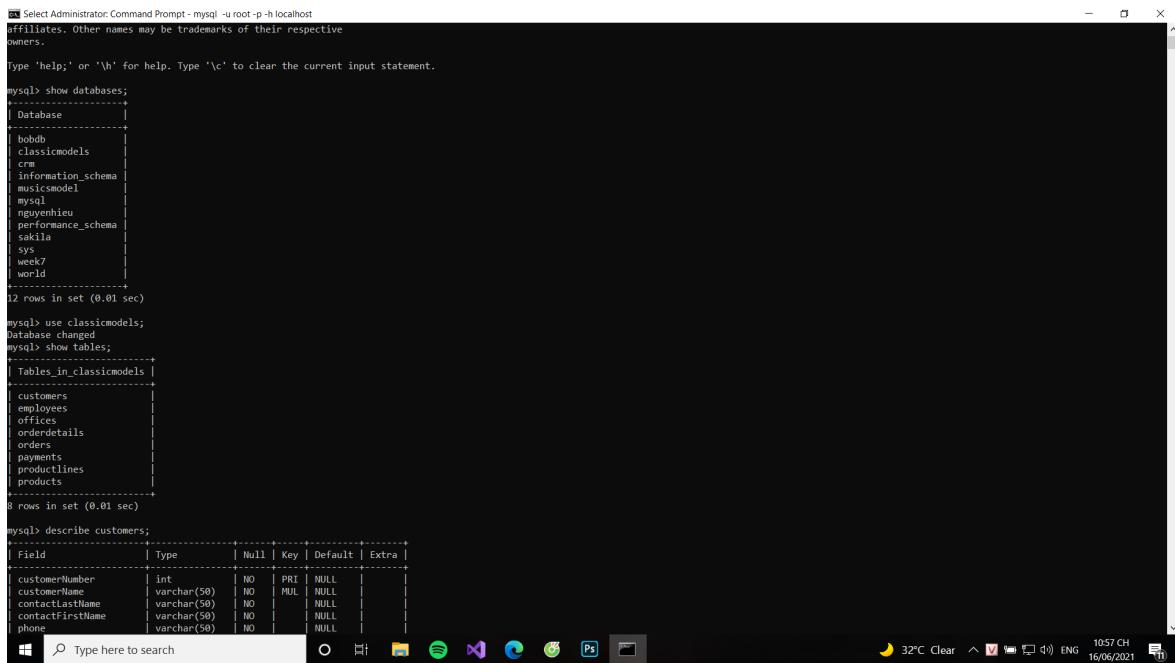
Hình 1.18: Bài 2 ([link](#))

1.9.3 Bài 3

Show commands

SOURCE CODE:

- Show databases
- Show tables

DISPLAY RESULTS:


```

Select Administrator: Command Prompt - mysql -u root -p -h localhost
Affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show databases;
+-----+
| Database |
+-----+
| bobdb   |
| classicmodels |
| crs     |
| information_schema |
| musicmodel |
| mysql    |
| nguyenvieul |
| performance_schema |
| sakila   |
| sys      |
| testk7   |
| world    |
+-----+
12 rows in set (0.01 sec)

mysql> use classicmodels;
Database changed
mysql> show tables;
+-----+
| Tables_in_classicmodels |
+-----+
| customers   |
| employees   |
| offices     |
| orderdetails |
| orders      |
| payments    |
| productlines |
| products    |
+-----+
8 rows in set (0.01 sec)

mysql> describe customers;
+-----+-----+-----+-----+-----+-----+
| Field        | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| customerNumber | int       | NO  | PRI | NULL    |       |
| customerName  | varchar(50) | NO  | MUL | NULL    |       |
| contactLastName | varchar(50) | NO  |     | NULL    |       |
| contactFirstName | varchar(50) | NO  |     | NULL    |       |
| phone         | varchar(50) | NO  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+

```

1.9.4 Bài 4*Backup and Restore***SOURCE CODE:**

- Backup: mysqldump –user=root –password=Super!rPass1 –result-file=c:\backup\classicmodels
- Restore: mysql>source c:\backup\mydb.sql

DISPLAY RESULTS:

1.9.5 Bài 5

Database maintenance

SOURCE CODE:

```
ANALYZE TABLE payments;  
OPTIMIZE TABLE payments;  
REPAIR TABLE payments;
```

DISPLAY RESULTS:

```
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 40
Server version: 8.0.23 MySQL Community Server - GPL

Copyright (c) 2000, 2021, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help'; or '\h' for help. Type '\c' to clear the current input statement.

mysql> analyze table customers;
ERROR 1046 (3D000): No database selected
mysql> use classicmodels;
Database changed
mysql> analyze table customers;
+-----+-----+-----+
| Table | Op   | Msg_type | Msg_text |
+-----+-----+-----+
| classicmodels.customers | analyze | status   | OK      |
+-----+-----+-----+
1 row in set (0.02 sec)

mysql> optimize table orders;
+-----+-----+-----+
| Table | Op   | Msg_type | Msg_text |
+-----+-----+-----+
| classicmodels.orders | optimize | note    | Table does not support optimize, doing recreate + analyze instead |
| classicmodels.orders | optimize | status   | OK      |
+-----+-----+-----+
2 rows in set (0.19 sec)

mysql> check table payments;
+-----+-----+-----+
| Table | Op   | Msg_type | Msg_text |
+-----+-----+-----+
| classicmodels.payments | check  | status   | OK      |
+-----+-----+-----+
1 row in set (0.01 sec)

mysql> repair table orders;
+-----+-----+-----+
| Table | Op   | Msg_type | Msg_text |
+-----+-----+-----+
| classicmodels.orders | repair | note    | The storage engine for the table doesn't support repair |
+-----+-----+-----+
1 row in set (0.01 sec)

mysql>
```

Chương 2

Kết luận

Qua 18 tuần học bộ môn Cơ Sở Dữ Liệu với sự giảng dạy nhiệt tình, tận tâm của Ths. Nguyễn Danh Tú, em đã nắm được những điều cơ bản nhất trong bộ môn này. Sau khi học xong em xin được tóm tắt những kiến thức đã được em cô đọng lại:

- Tư duy thiết kế CSDL, đọc và phân tích các lược đồ.
- Hiểu được mối quan hệ giữa các thực thể trong lược đồ, ý nghĩa của những dữ liệu. Từ đó có thể phát triển, nâng cấp hay chuẩn hóa các CSDL.
- Cách đặt câu hỏi để khai phá dữ liệu có trong CSDL, nắm được các đặc trưng của CSDL và đưa ra những truy vấn chính xác.
- Sử dụng được các hệ quản trị CSDL: tạo CSDL, thay đổi cấu trúc CSDL, nâng cấp và bảo hành CSDL.
- Cách xử lí các dữ liệu trong khi đưa vào CSDL.
- Viết được các procedure và functions từ cơ bản đến phức tạp.
- Tối ưu các truy vấn bằng các phương pháp như tinh chỉnh hay tạo index, partition.
- Hiểu và thực thi được các nghiệp vụ thực tế liên quan đến CSDL.
- Nắm được các định nghĩa, tính chất, thuật toán: Sự bất thường trong CSDL, các tính chất của phụ thuộc hàm, tiên đề Armstrong và hệ quả, các phương pháp tìm khóa, cách tách lược đồ, kiểm tra phép tách, các dạng chuẩn CSDL.
- Làm được các tác vụ nâng cao liên quan đến Administration.

- Nâng cao năng suất làm việc với các thành viên trong nhóm, giao tiếp để đạt hiệu quả cao, cùng với kỹ năng thuyết trình và viết báo cáo.

Thời gian 18 tuần cũng không phải là quá dài nên trong quá trình học có nhiều chỗ khiến em chưa hiểu và gặp khó khăn, một số nội kiến thức bị bỏ sót nên em mong bản báo cáo này sẽ là một cách để em rà soát lại những kiến thức. Em xin cảm ơn những kiến thức bổ ích mà thầy **Nguyễn Danh Tú** đã giảng dạy cho lớp trong học phần này.

Tài liệu tham khảo

- [1] Google.
- [2] MySQL Documentation.
- [3] viblo.asia
- [4] Slide.