**Bài thực hành 03 – Tuần 8, ngày 16 tháng 11 năm 2021**

Contents

[5. BÀI THỰC HÀNH SỐ 03 – THIẾT KẾ LỚP VÀ MÔ HÌNH HÓA DỮ LIỆU 2](#_Toc88235847)

[5.3. NỘI DUNG CHI TIẾT 2](#_Toc88235848)

[5.3.1. Thiết kế lớp (Class Design) 2](#_Toc88235849)

[5.3.2. Mô hình hóa dữ liệu (Data Modeling) 13](#_Toc88235850)

[5.4. BÀI TẬP 21](#_Toc88235851)

[5.4.1. Thiết kế lớp 21](#_Toc88235852)

[5.4.2. Mô hình hóa dữ liệu 22](#_Toc88235853)

- Link đã nộp bài lên Github:

**https://github.com/ductuan-nguyen/TKXDPM.KSCQ.20211.20180192.NguyenDucTuan**

- Nội dung báo cáo kết quả thực hành

# 5. BÀI THỰC HÀNH SỐ 03 – THIẾT KẾ LỚP VÀ MÔ HÌNH HÓA DỮ LIỆU

## 5.3. NỘI DUNG CHI TIẾT

### 5.3.1. Thiết kế lớp (Class Design)

#### 5.3.1.1. Bước đầu tạo các lớp thiết kế

Text

Description automatically generatedText

Description automatically generated

Graphical user interface, text, application, chat or text message

Description automatically generated

#### 5.3.1.2. Xác định mối quan hệ giữa các lớp

##### a) Relationship Exception Class Diagram

Diagram

Description automatically generated

##### b) Relationship class diagram

Diagram

Description automatically generated

#### 5.3.1.3. Lớp thiết kế

##### a) Lớp “InterbankInterface”

Text, letter

Description automatically generated

***Attribute***

Không

***Operation***

|  |  |  |  |
| --- | --- | --- | --- |
| *#* | *Tên* | *Kiểu dữ liệu trả về* | *Mô tả (mục đích)* |
| 1 | payOrder | PaymentTransaction | Thanh toán đơn hàng và trả về giao dịch thanh toán |
| 2 | refund | PaymentTransaction | Hoàn tiền và trả về giao dịch thanh toán |
| 3 | getBalance | int | Trả về số dư tài khoản |

*Parameter:*

* card – thẻ tín dụng để giao dịch
* amount – số tiền giao dịch
* contents – nội dung giao dịch

*Exception:*

* PaymentException – nếu mã lỗi trả về đã biết
* UnrecognizedException – nếu không tìm thấy mã lỗi trả về hoặc có lỗi hệ thống

##### b) Lớp “PaymentController”

Table

Description automatically generated

***Attribute***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *#* | *Tên* | *Kiểu dữ liệu trả về* | *Giá trị mặc định* | *Mô tả* |
| 1 | card | CreditCard | NULL | Represent the card used for payment |
| 2 | interbank | InterbankInterface | NULL | Represent the Interbank subsystem |

***Operation***

|  |  |  |  |
| --- | --- | --- | --- |
| *#* | *Tên* | *Kiểu dữ liệu trả về* | *Mô tả (mục đích)* |
| 1 | payOrder | Map<String,String> | Thanh toán đơn hàng và trả về giao dịch thanh toán |

*Parameter:*

* amount – số tiền giao dịch
* contents – nội dung giao dịch
* cardNumber – số thẻ
* cardHolderName – tên chủ sở hữu
* expirationDate – ngày hết hạn theo định dạng "mm/yy"
* securityCode - mã bảo mật cvv/cvc

*Exception:*

* Không

***Method***

* getExpirationDate: Chuyển dữ liệu ngày từ định dạng “mm/yy” sang “mmyy”.

***Diagram

Description automatically generated***

***State***

Không

##### c) Biểu đồ trạng thái (state machine) cho đối tượng “Order”

Diagram

Description automatically generated

##### d) Lớp “PlaceOrderController”

Table

Description automatically generated

***Attribute:***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *#* | *Tên* | *Kiểu dữ liệu trả về* | *Giá trị mặc định* | *Mô tả* |
| 1 | cart | Cart | NULL | Giỏ hàng của người dùng |
| 2 | invoice | Invoice | NULL | Hóa đơn đặt hàng |
| 3 | order | Order | NULL | Đơn hàng |

***Operation:***

|  |  |  |  |
| --- | --- | --- | --- |
| *#* | *Tên* | *Kiểu dữ liệu trả về* | *Mô tả (mục đích)* |
| 1 | placeOrder | Invoice | Đăt hàng, khởi tạo đơn hàng, tạo và trả về thông tin hóa đơn |
| 2 | proccessDeliveryInfo | boolean | Kiểm tra thông tin giao hàng |
| 3 | confirmInvoice | void | Lưu thông tin đặt hàng |
| 4 | proccessRushDeliveryInfo | boolean | Kiểm tra thông tin giao hàng nhanh |

*Parameters:*

* cart – giỏ hàng
* info – thông tin giao hàng

*Exception:*

* InvalidDeliveryInfoException – thông tin giao hàng không hợp lệ
* InvalidRushDeliveryInfoException – thông tin giao hàng nhanh không hợp lệ
* PlaceOrderException – lỗi đặt hàng

***Method:***

* validateDeliveryInfo: Kiểm tra thông tin giao hàng, kiểm tra các trường bắt buộc, kiểm tra định dạng
* caculateShippingFee: Tính toán phí giao hàng, lưu vào Invoice

***State:*** Không

##### e) Lớp “ViewCartController”

Table

Description automatically generated with medium confidence

***Attribute:***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *#* | *Tên* | *Kiểu dữ liệu trả về* | *Giá trị mặc định* | *Mô tả* |
| 1 | cart | Cart | NULL | Giỏ hàng của người dùng |

***Operation:***

|  |  |  |  |
| --- | --- | --- | --- |
| *#* | *Tên* | *Kiểu dữ liệu trả về* | *Mô tả (mục đích)* |
| 1 | checkAvailabilityOfProduct | bool | Kiểm tra các sản phaame trong giỏ hàng có còn đủ số lượng hay không |
| 2 | getCartInfo | Cart | Lấy thông tin giỏ hàng |
| 3 | updateCart | void | Cập nhập giỏ hàng |

*Parameter:* Không

*Exception:*

* ViewCartException: Lỗi xem giỏ hàng

***Method:*** Không

##### f) Lớp “PaymentScreenHandler”

Graphical user interface, text, application

Description automatically generated

***Attribute:***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *#* | *Tên* | *Kiểu dữ liệu trả về* | *Giá trị mặc định* | *Mô tả* |
| 1 | paymentController | PaymentController | NULL | Controller xử lý nghiệp vụ thanh toán |

***Operation:***

|  |  |  |  |
| --- | --- | --- | --- |
| *#* | *Tên* | *Kiểu dữ liệu trả về* | *Mô tả (mục đích)* |
| 1 | requestToPayOrder | void | Hiển thị màn hình thanh toán |
| 2 | confirmToPayOrder | void | Thực hiện thanh toán |
| 3 | notifyError | void | Thông báo lỗi |

*Parameters:*

* amount – số tiền giao dịch
* contents – nội dung giao dịch
* cardNumber – số thẻ
* cardHolderName – tên chủ sở hữu
* expirationDate – ngày hết hạn theo định dạng "mm/yy"
* securityCode - mã bảo mật cvv/cvc
* invoice – thông tin hóa đơn
* message – thông báo lỗi

*Exception:* Không

***Method:*** Không

##### g) Lớp “InvoiceScreen”

Table

Description automatically generated  
  
***Attribute:***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *#* | *Tên* | *Kiểu dữ liệu trả về* | *Giá trị mặc định* | *Mô tả* |
| 1 | invoice | Invoice | NULL | Thông tin hóa đơn thanh toán |

***Operation:***

|  |  |  |  |
| --- | --- | --- | --- |
| *#* | *Tên* | *Kiểu dữ liệu trả về* | *Mô tả (mục đích)* |
| 1 | displayInvoice | void | HIển thị màn hình thông tin hóa đơn |
| 2 | confirmInvoice | void | Đồng ý và hiển thị màn hình thanh toán |
| 3 | displaySuccessOrder | void | HIển thị đặt hàng thành công sau khi thanh toán |

*Parameters:*

* Invoice – thông tin hóa đơn thanh toán

*Exception:* Không

***Method:*** Không

#### 5.3.1.4. Biểu đồ lớp thiết kế

Diagram

Description automatically generated

### 5.3.2. Mô hình hóa dữ liệu (Data Modeling)

#### 5.3.2.1. Mô hình khái niệm

Diagram

Description automatically generated

#### 5.3.2.2 Thiết kế cơ sở dữ liệu

##### a) Mô hình dữ liệu logic

Diagram

Description automatically generated

##### b) Mô hình dữ liệu vật lý

**Media**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| # | PK | FK | Tên cột | Kiểu dữ liệu | Bắt buộc | Mô tả |
| 1 | x |  | id | Integer | Có | ID, auto increment |
| 2 |  |  | category | VARCHAR(45) | Có | Thể loại, e.g., CD, DVD |
| 3 |  |  | price | Integer | Có | Giá hiện tại |
| 4 |  |  | quantity | Integer | Có | Số lượng sản phẩm |
| 5 |  |  | title | VARCHAR(45) | Có | Tên sản phẩm |
| 6 |  |  | value | Integer | Có | Giá trị sản phẩm |
| 7 |  |  | imageUrl | VARCHAR(45) | Có | Đường dẫn hình ảnh của sản phẩm |

**CD**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| # | PK | FK | Tên cột | Kiểu dữ liệu | Bắt buộc | Mô tả |
| 1 |  | x | id | Integer | Có | ID của CD |
| 2 |  |  | artist | VARCHAR(45) | Có | Tên nghệ sĩ |
| 3 |  |  | recordLabel | VARCHAR(45) | Có | Nhãn bản ghi |
| 4 |  |  | musicType | VARCHAR(45) | Có | Thể loại nhạc |
| 5 |  |  | releasedDate | DATE | Có | Ngày phát hành |

**Book**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| # | PK | FK | Tên cột | Kiểu dữ liệu | Bắt buộc | Mô tả |
| 1 |  | x | id | Integer | Có | ID của sách |
| 2 |  |  | author | VARCHAR(45) | Có | Tác giả |
| 3 |  |  | coverType | VARCHAR(45) | Có | Loại bìa |
| 4 |  |  | Publisher | VARCHAR(45) | Có | Nhà xuất bản |
| 5 |  |  | publishDate | DATETIME | Có | Ngày xuất bản |
| 6 |  |  | numOfPages | Integer | Có | Số trang |
| 7 |  |  | language | VARCHAR(45) | Có | Ngôn ngữ |
| 8 |  |  | bookCategory | VARCHAR(45) | Có | Thể loại sách |

**DVD**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| # | PK | FK | Tên cột | Kiểu dữ liệu | Bắt buộc | Mô tả |
| 1 |  | x | id | Integer | Có | ID của DVD |
| 2 |  |  | discType | VARCHAR(45) | Có | Loại đĩa |
| 3 |  |  | director | VARCHAR(45) | Có | Đạo diễn |
| 4 |  |  | runtime | Integer | Có | Thời lượng |
| 5 |  |  | studio | VARCHAR(45) | Có | Nhà sản xuất |
| 6 |  |  | subtitle | VARCHAR(45) | Có | Phụ đề |
| 7 |  |  | releasedDate | DATETIME | Có | Ngày phát hành |
| 8 |  |  | filmType | VARCHAR(45) | Có | Thể loại |

**Card**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| # | PK | FK | Tên cột | Kiểu dữ liệu | Bắt buộc | Mô tả |
| 1 | x |  | id | Integer | Có | ID tự động tăng |
| 2 |  |  | cardCode | VARCHAR(45) | Có | Mã thẻ |
| 3 |  |  | owner | VARCHAR(45) | Có | Người sở hữu |
| 4 |  |  | cvvCode | VARCHAR(3) | Có | Mã CVV |
| 5 |  |  | dateExpired | VARCHAR(4) | Có | Ngày hết hạn |

**DeliveryInfo**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| # | PK | FK | Tên cột | Kiểu dữ liệu | Bắt buộc | Mô tả |
| 1 | X |  | id | Integer | Có | ID, tự động tăng |
| 2 |  |  | name | VARCHAR(45) | Có | Tên người nhận |
| 3 |  |  | province | VARCHAR(45) | Có | Tỉnh |
| 4 |  |  | instructions | VARCHAR(200) | Không | Hướng dẫn vận chuyển |
| 5 |  |  | address | VARCHAR(100) | Có | Địa chỉ vận chuyển |

**Order**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| # | PK | FK | Tên cột | Kiểu dữ liệu | Bắt buộc | Mô tả |
| 1 | X |  | id | Integer | Có | ID |
| 2 |  |  | shippingFees | VARCHAR(45) | Có | Phí vận chuyển |
| 3 |  | X | deliveryInfoId | Integer | Có | ID của DeliveryInfo |

**Order Media**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| # | PK | FK | Tên cột | Kiểu dữ liệu | Bắt buộc | Mô tả |
| 1 |  | X | mediaID | Integer | Có | ID của Media |
| 2 |  | X | orderID | Integer | Có | ID của Order |
| 3 |  |  | price | Integer | Có | Giá bán |
| 4 |  |  | quantity | Integer | Có | Số lượng |

**Invoice**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| # | PK | FK | Tên cột | Kiểu dữ liệu | Bắt buộc | Mô tả |
| 1 | x |  | id | Integer | Có | ID |
| 2 |  |  | totalAmount | Integer | Có | Tổng số lượng |
| 3 |  | x | orderId | Integer | Có | ID của order |

**PaymentTransaction**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| # | PK | FK | Tên cột | Kiểu dữ liệu | Bắt buộc | Mô tả |
| 1 | x |  | id | Integer | Yes | ID |
| 2 |  |  | createAt | DATETIME | Yes | Ngày tạo |
| 3 |  |  | content | VARCHAR(45) | Yes | Nội dung giao dịch |
| 4 |  |  | method | VARCHAR(45) | Yes | Phương thức thanh toán |
| 5 |  | x | cardId | Integer | Yes | ID của card |
| 6 |  | x | invoiceId | Integer | Yes | ID của invoice |

**Database scripts:**

BEGIN;

create table aims. mediaTuanND

(

id INTEGER PRIMARY KEY auto\_increment NOT NULL ,

category varchar(45) not null,

price INTEGER not null,

quantity INTEGER not null,

title varchar(45) not null,

value INTEGER not null,

imageUrl varchar(45) not null

);

create table aims. CDTuanND

(

id INTEGER PRIMARY KEY not null,

artist VARCHAR(45) not null,

recordLabel VARCHAR(45) not null,

musicType VARCHAR(45) not null,

releasedDate DATE,

constraint fk\_CD\_Media1

foreign key (id) references mediaTuanND (id)

);

create table aims. BookTuanND

(

id INTEGER PRIMARY KEY not null ,

author VARCHAR(45) not null ,

coverType VARCHAR(45) not null ,

publisher VARCHAR(45) not null ,

publishDate DATETIME not null ,

numOfPages INTEGER not null ,

language VARCHAR(45) not null ,

bookCategory VARCHAR(45) not null ,

constraint fk\_Book\_Media1

foreign key (id) references mediaTuanND (id)

);

create table aims. DeliveryInfoTuanND

(

id INTEGER PRIMARY KEY AUTO\_INCREMENT not null ,

name VARCHAR(45),

province VARCHAR(45),

instructions VARCHAR(200),

address VARCHAR(100),

isRushOrder boolean

);

create table aims. CardTuanND

(

id INTEGER PRIMARY KEY AUTO\_INCREMENT not null ,

cardCode VARCHAR(15) not null ,

owner VARCHAR(45) not null ,

cvvCode VARCHAR(3) not null ,

dateExpired VARCHAR(4) not null

);

create table aims. DVDTuanND

(

id INTEGER primary key not null ,

discType VARCHAR(45) not null ,

director VARCHAR(45) not null ,

runtime INTEGER not null ,

studio VARCHAR(45) not null ,

subtitle VARCHAR(45) not null ,

releasedDate DATETIME,

constraint fk\_DVD\_Media1

foreign key (id) references mediaTuanND (id)

);

create table aims. OrderTuanND

(

id INTEGER not null ,

shippingFees VARCHAR(45) ,

deliveryInfoId INTEGER not null ,

primary key (id, deliveryInfoId) ,

constraint fk\_Order\_DeliveryInfo1

foreign key (deliveryInfoId) references DeliveryInfoTuanND (id)

);

create index fk\_Order\_DeliveryInfo1\_idx ON OrderTuanND (deliveryInfoId);

create table aims. OrderMediaTuanND(

orderID INTEGER not null ,

price INTEGER not null ,

quantity INTEGER not null ,

mediaId INTEGER not null ,

primary key (orderID, mediaId),

constraint fk\_ordermedia\_order foreign key (orderID) references OrderTuanND(id),

constraint fk\_OrderMedia\_Media1 foreign key (mediaId) references mediaTuanND (id)

);

create index fk\_ordermedia\_order\_idx ON OrderMediaTuanND (orderID);

create index fk\_OrderMedia\_Media1\_idx ON OrderMediaTuanND (mediaId);

create table aims. InvoiceTuanND

(

id INTEGER PRIMARY KEY not null ,

totalAmount INTEGER not null ,

orderId INTEGER not null ,

constraint fk\_Invoice\_Order1

foreign key (orderId) references OrderTuanND(id)

);

create index fk\_Invoice\_Order1\_idx ON InvoiceTuanND (orderId);

create table aims. PaymentTransactionTuanND(

id INTEGER not null ,

createAt DATETIME not null ,

content VARCHAR(45) not null ,

method VARCHAR(45),

cardId INTEGER not null ,

invoiceId INTEGER not null ,

primary key (id, cardId, invoiceId),

constraint fk\_PaymentTransaction\_Card1 foreign key (cardId) references CardTuanND(id),

constraint fk\_PaymentTransaction\_Invoice1 foreign key (invoiceId) references InvoiceTuanND(id)

);

create index fk\_PaymentTransaction\_Card1\_idx ON PaymentTransactionTuanND (cardId);

create index fk\_PaymentTransaction\_Invoice1\_idx ON PaymentTransactionTuanND (invoiceId);

COMMIT;

## 5.4. BÀI TẬP

### 5.4.1. Thiết kế lớp

#### 5.4.1.1. Mối quan hệ giữa các lớp

Bổ sung lớp Place Rush Order Controller ta có kết quả sau:

Diagram

Description automatically generated

#### 5.4.1.2. Lớp thiết kế

##### Lớp “PlaceRushOrderController”

Text

Description automatically generated with low confidence

***Attribute:***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *#* | *Tên* | *Kiểu dữ liệu trả về* | *Giá trị mặc định* | *Mô tả* |
| 1 | cart | Cart | NULL | Giỏ hàng của người dùng |

***Operation:***

|  |  |  |  |
| --- | --- | --- | --- |
| *#* | *Tên* | *Kiểu dữ liệu trả về* | *Mô tả (mục đích)* |
| 1 | placeRushOrder | Invoice | Đặt đơn hàng giao nhanh |
| 2 | processRushDeliveryInfo | Cart | Xử lý thông tin đơn hàng giao nhanh |
| 3 | validateRushDeliveryInfo | boolean | Kiểm tra thông tin đơn hàng giao nhanh |

*Parameters:*

Cart: thông tin giỏ hàng

rushTime: thời gian giao hàng nhanh

rushAddress: địa điểm giao hàng nhanh

rushInstruction: hướng dẫn giao hàng nhanh

*Exception:*

* CartException: Lỗi xem giỏ hàng

***Method:*** Không

### 5.4.2. Mô hình hóa dữ liệu

#### 5.4.2.1. Mô hình khái niệm

Thêm trường thông tin isRushOrder cho bảng DeliveryInfo và tạo mới bảng RushOrder ta có kết quả sau:

Diagram

Description automatically generated

#### 5.4.2.2. Thiết kế cơ sở dữ liệu

##### a) Mô hình dữ liệu logic

Bảng dữ liệu Delivery Info có thêm trường isRushOrder

Tạo thêm bảng dữ liệu rushorder để lưu trữ thông tin các đơn giao hàng nhanh (Rush Order)

Các bảng còn lại không thay đổi so với mô hình dữ liệu logic của Place Order

Kết quả mô hình dữ liệu logic như sau:

Diagram

Description automatically generated

##### b) Mô hình dữ liệu vật lý

- Các bảng dữ liệu sau đây có sự thay đổi:

**DeliveryInfo**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| # | PK | FK | Tên cột | Kiểu dữ liệu | Bắt buộc | Mô tả |
| 1 | X |  | id | Integer | Có | ID, tự động tăng |
| 2 |  |  | name | VARCHAR(45) | Có | Tên người nhận |
| 3 |  |  | province | VARCHAR(45) | Có | Tỉnh |
| 4 |  |  | instructions | VARCHAR(200) | Không | Hướng dẫn vận chuyển |
| 5 |  |  | address | VARCHAR(100) | Có | Địa chỉ vận chuyển |
| 6 |  |  | isRushOrder | TINYINT | Có | Có sử dùng dịch vụ giao hàng nhanh hay không |

- Tạo mới bảng sau:

**RushOrder**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| # | PK | FK | Tên cột | Kiểu dữ liệu | Bắt buộc | Mô tả |
| 1 | X |  | id | Integer | Có | ID |
| 2 |  |  | rushDeliveryTime | Datetime | Có | Thời gian giao hàng nhanh |
| 3 |  |  | rushAddress | String | Có | Địa điểm giao hàng nhanh |
| 4 |  |  | rushInstruction | String | Có | Hướng dẫn giao hàng nhanh |
| 5 |  | X | orderId | Integer | Có | ID của order tương ứng |

- Các bảng còn lại không thay đổi so với UC Place Order

**Database script:**

BEGIN;

create table aims. mediaTuanND

(

id INTEGER PRIMARY KEY auto\_increment NOT NULL ,

category varchar(45) not null,

price INTEGER not null,

quantity INTEGER not null,

title varchar(45) not null,

value INTEGER not null,

imageUrl varchar(45) not null

);

create table aims. CDTuanND

(

id INTEGER PRIMARY KEY not null,

artist VARCHAR(45) not null,

recordLabel VARCHAR(45) not null,

musicType VARCHAR(45) not null,

releasedDate DATE,

constraint fk\_CD\_Media1

foreign key (id) references mediaTuanND (id)

);

create table aims. BookTuanND

(

id INTEGER PRIMARY KEY not null ,

author VARCHAR(45) not null ,

coverType VARCHAR(45) not null ,

publisher VARCHAR(45) not null ,

publishDate DATETIME not null ,

numOfPages INTEGER not null ,

language VARCHAR(45) not null ,

bookCategory VARCHAR(45) not null ,

constraint fk\_Book\_Media1

foreign key (id) references mediaTuanND (id)

);

create table aims. DeliveryInfoTuanND

(

id INTEGER PRIMARY KEY AUTO\_INCREMENT not null ,

name VARCHAR(45),

province VARCHAR(45),

instructions VARCHAR(200),

address VARCHAR(100),

isRushOrder boolean

);

create table aims. CardTuanND

(

id INTEGER PRIMARY KEY AUTO\_INCREMENT not null ,

cardCode VARCHAR(15) not null ,

owner VARCHAR(45) not null ,

cvvCode VARCHAR(3) not null ,

dateExpired VARCHAR(4) not null

);

create table aims. DVDTuanND

(

id INTEGER primary key not null ,

discType VARCHAR(45) not null ,

director VARCHAR(45) not null ,

runtime INTEGER not null ,

studio VARCHAR(45) not null ,

subtitle VARCHAR(45) not null ,

releasedDate DATETIME,

constraint fk\_DVD\_Media1

foreign key (id) references mediaTuanND (id)

);

create table aims. OrderTuanND

(

id INTEGER not null ,

shippingFees VARCHAR(45) ,

deliveryInfoId INTEGER not null ,

primary key (id, deliveryInfoId) ,

constraint fk\_Order\_DeliveryInfo1

foreign key (deliveryInfoId) references DeliveryInfoTuanND (id)

);

create index fk\_Order\_DeliveryInfo1\_idx ON OrderTuanND (deliveryInfoId);

create table aims. OrderMediaTuanND(

orderID INTEGER not null ,

price INTEGER not null ,

quantity INTEGER not null ,

mediaId INTEGER not null ,

primary key (orderID, mediaId),

constraint fk\_ordermedia\_order foreign key (orderID) references OrderTuanND(id),

constraint fk\_OrderMedia\_Media1 foreign key (mediaId) references mediaTuanND (id)

);

create index fk\_ordermedia\_order\_idx ON OrderMediaTuanND (orderID);

create index fk\_OrderMedia\_Media1\_idx ON OrderMediaTuanND (mediaId);

create table aims. InvoiceTuanND

(

id INTEGER PRIMARY KEY not null ,

totalAmount INTEGER not null ,

orderId INTEGER not null ,

constraint fk\_Invoice\_Order1

foreign key (orderId) references OrderTuanND(id)

);

create index fk\_Invoice\_Order1\_idx ON InvoiceTuanND (orderId);

create table aims. PaymentTransactionTuanND(

id INTEGER not null ,

createAt DATETIME not null ,

content VARCHAR(45) not null ,

method VARCHAR(45),

cardId INTEGER not null ,

invoiceId INTEGER not null ,

primary key (id, cardId, invoiceId),

constraint fk\_PaymentTransaction\_Card1 foreign key (cardId) references CardTuanND(id),

constraint fk\_PaymentTransaction\_Invoice1 foreign key (invoiceId) references InvoiceTuanND(id)

);

create index fk\_PaymentTransaction\_Card1\_idx ON PaymentTransactionTuanND (cardId);

create index fk\_PaymentTransaction\_Invoice1\_idx ON PaymentTransactionTuanND (invoiceId);

create table aims. RushOrderTuanND

(

id INTEGER not null ,

orderId INTEGER not null ,

rushDeliveryTime DATETIME not null ,

rushAddress VARCHAR(200) not null ,

rushInstruction VARCHAR(200) not null ,

primary key (id, orderId),

constraint fk\_RushOrder\_Order

foreign key (id) references OrderTuanND(id)

);

create index fk\_RushOrder\_Order\_idx ON RushOrderTuanND (orderId);

COMMIT;