Database Coursework 1 Group Report
Group TWELVE

Sakila

Group Members

Kok Yong En	10346769
Lim Hui Xing	20014709
Lim Kelly	20024694
Sin Kai Ling	10341812
Tan Song Ning	10281222
Tan Yi-Wei	20033760

Introduction

The dataset Sakila is a group of data related to a film rental shop. The dataset is given with its entities and attributes defined. The relation for all the entities with their attributes have been provided as well.

Reason behind the design of database

Database allows us to store information, and provides us efficiency in manipulating the data, which includes organizing, retrieving, updating the data.

One of the major reasons behind this database design is to reduce redundancy and inconsistency. Without a proper design of database, duplicated information could exist in several places (files). Redundancy will lead to higher cost of storage and access cost. If the various copies of information does not share the same information, inconsistencies arise.

Besides that, with the relationships between relations clearly identified (primary keys and foreign keys), integrity problems could be avoided. The connections between the tables enables us to put constraints on to the information. For our database Sakila, foreign keys that exist in the table `payment`, are `SET NULL` on delete, but the foreign keys in other tables such as `film, customers, address, payment` etc. `CASCADE` on delete and on update. This is because the records on payment are meant to be kept even if the referenced entities no longer exist.

Connection between the Tables
The primary keys and the foreign keys for the dataset are identified.

Candidate keys	Primary key in the relation	Foreign key in the relation
actor_id	actor	film_actor
address_id	address	customer, store, staff
city_id	city	address
country_id	country	city
customer_id	customer	payment, rental

film_id	film	film_actor, inventory
language_id	language	film
inventory_id	inventory	rental
payment_id	payment	
rental_id	rental	payment
store_id	store	customer, inventory, staff
staff_id	staff	payment, rental

Data types

Primary keys are usually in the form of ID. The most convenient data type for ID is integer types. Thus, for numerics data with lesser than 3 digits, the data type `tinyint` is used. For all the fields that are related to numerics which have lesser than 5 digits, the data type `smallint` is used. For numerics data with lesser than 8 digits, the data type `mediumint` is used. For numerics data with lesser than 11 digits, the data type `int` is used. Fields like 'active' which have boolean values like True(1) or False(0) use data type `tinyint` as well.

The fields which are related to names, for instance, customer names, street names, city names, country names, category names or usernames uses the data type `varchar`. Other fields such as passwords, email and descriptions for the film also use the data type `varchar`. Varchar means a variable character field which can hold characters and numerics. However, the field description is assigned the data type `text`. This is due to descriptions are usually lengthy and that data type can hold up to 65535 characters.

Fields which are associated with prices such as amount and cost uses the data type `decimal`. These fields are floating point numbers are the precision up until two decimal points.

Fields which records the time and date about the creation time or its last update use the data type `timestamp`. This data type supports the format of YYYY-MM-DD HH:MM:SS.

The field 'rating' in the table 'film' uses the data type `enum,` as the ratings for each film can only be 'G', 'PG', 'PG-13', 'R', or 'NC-17'. Only one option in the set can be used.

The field 'special features' in the table 'film' uses the datatype `Set,` as special features can be one or more options from "Trailers, Deleted Scenes, Commentaries, Behind the Scenes".

Relationships

Every film is acted by multiple actors, every actor is involved in many films. Thus, entity actor and entity film have a many-to-many relationship.

There are several copies of one same film in the inventory, each of the copies are given a unique ID. Thus, film and inventory share a one-to-many relationship.

There are many films which use the same language, but each film uses one language only. Language and film share a one-to-many relationship.

Each category has multiple film, every film has only one category. Category and film has a one-to-many relationship.

Every customer has only one address. Every staff has one address. Every store has one address. Every address is only owned by one entity. Thus, customer, staff, store share a one-to-one relationship with address.

Each city contains many addresses, each addresses only exist in one city. City and address have a one-to-many relationship.
Each country contains many cities, every city is situated in one country only. Thus country and city have a one-to-many relationship.

Each customer has made one or more payments, each payment can only be made by one customer.Customer and payment have a one-to-many relationship.

Each customer has made many rentals, each rental is only made by one customer. Customer and rental have a one-to-many relationship.

Every copy of film in the inventory can be rented one or multiple times. Each rental ID is associated with one copy of the film in the inventory only(inventory ID). Thus, inventory and rental have a one-to-many relationship.

Every rental has one or many payments. Each payment is made for one rental only. Rental and payment have a one-to-many relationship.

Every store is visited by many customers, every customer visits only one store. Store and customer have a one-to-one relationship.

Every store has multiple inventory ID, but each inventory ID is kept in one store only. Store and inventory have a one-to-many relationship.

Every store has many staff, every staff works in one store only. Store and staff have a one-to-many relationship.

Every staff has managed many payments, each payment is handled by one staff only. Staff and payment have a one-to-many relationship.

Every staff has managed many rentals, every rental is handled by one staff only. Staff and rental have a one-to-many relationship.

Errors Found and Solutions

In staff relation, the table contains meaningless sequence of characters and numbers. The table also include a picture column which is not a supported data type in database management system. Therefore, all the unnecessary datas are removed from the relation.

There are data redundancies in table 'film_text' and table 'category', so both the tables are removed and a new column category_id is added into the table 'film'.

Completeness

There are some fields in the data set which are null, which are 'Address2' from 'address' table, 'original language id' from from 'film' table. There are also incompleteness in other entities like address entity which contains a few null data for phone and postal code attributes. It is assumed that there is some missing data from the data set.

ER Diagram
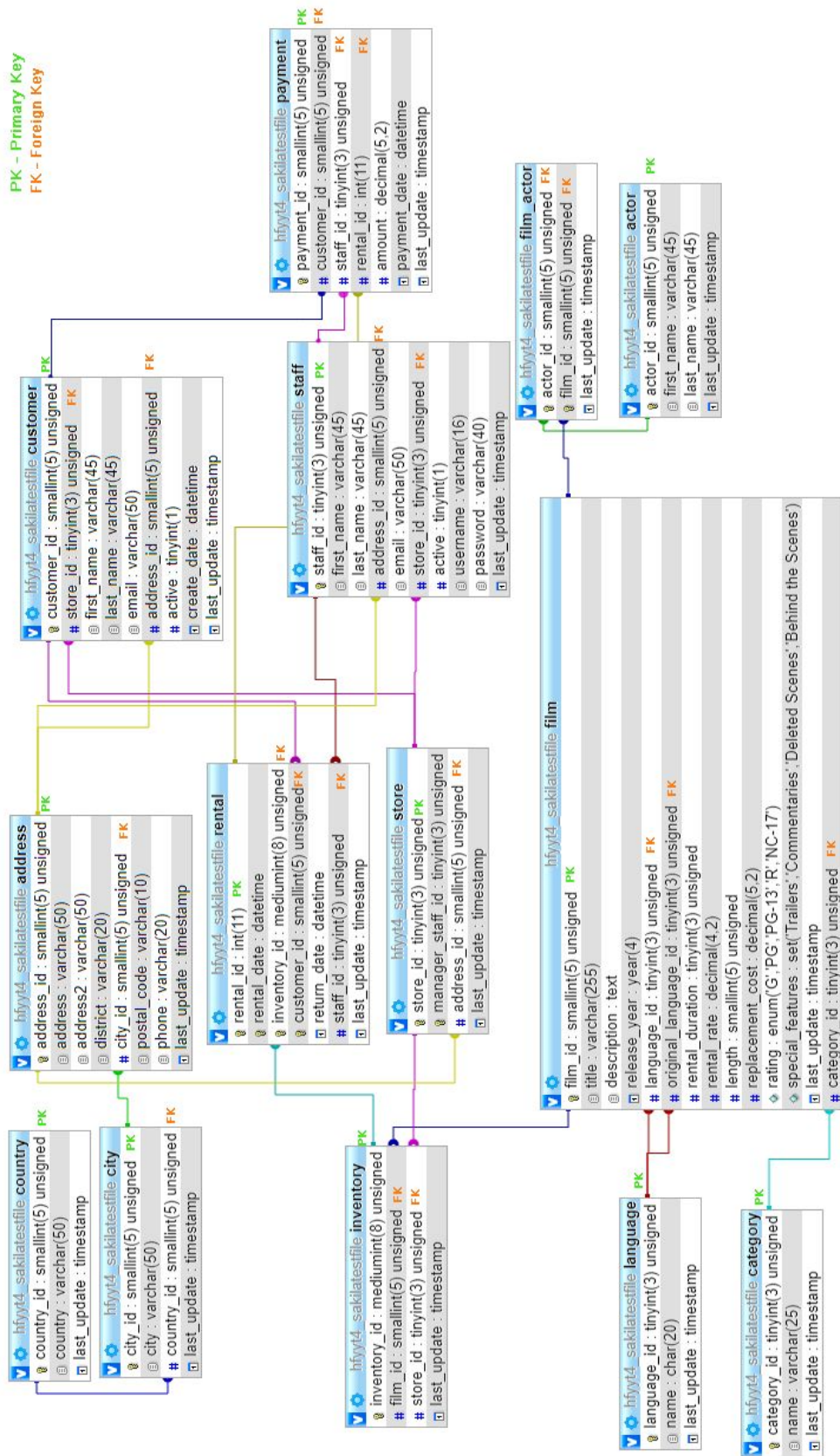
SQL Statements

Delete

DELETE FROM actor WHERE actor_id='$actor_id'

DELETE FROM address WHERE address_id='$address_id'

DELETE FROM category WHERE category_id='$category_id'

DELETE FROM city WHERE city_id='$city_id'

DELETE FROM country WHERE country_id='$country_id'

DELETE FROM customer WHERE customer_id='$customer_id'

DELETE FROM film WHERE film_id='$film_id'

DELETE FROM film_actor WHERE actor_id='$actor_id' AND film_id='$film_id'

DELETE FROM inventory WHERE inventory_id='$inventory_id'

DELETE FROM language WHERE language_id='$language_id'

DELETE FROM payment WHERE payment_id='$payment_id'

DELETE FROM rental WHERE rental_id='$rental_id'

DELETE FROM staff WHERE staff_id='$staff_id'

DELETE FROM store WHERE store_id='$store_id'

Insert

INSERT INTO actor
VALUES(NULL,'$first_name','$last_name',CURRENT_TIMESTAMP)

INSERT INTO address
VALUES(NULL,'$address','$address2','$district','$city_id','$postal_code','$phone',CURRENT_TIMESTAMP)

INSERT INTO category
VALUES(NULL,'$name',CURRENT_TIMESTAMP)

INSERT INTO city
VALUES(NULL,'$city','$country_id',CURRENT_TIMESTAMP)

INSERT INTO country
VALUES(NULL,'$country',CURRENT_TIMESTAMP)

INSERT INTO customer
VALUES(NULL,'$store_id','$first_name','$last_name','$email','$address_id','$active',CURRENT_TIMESTAMP,CURRENT_TIMESTAMP)

INSERT INTO film
VALUES(NULL,'$title','$description','$release_year','$language_id','$original_language_id','$rental_duration','$rental_rate','$length','$replacement_cost','$rating','$special_features',CURRENT_TIMESTAMP,'$category_id')
INSERT INTO film_actor
VALUES('$actor_id','$film_id',CURRENT_TIMESTAMP)
INSERT INTO inventory
VALUES(NULL,'$film_id','$store_id',CURRENT_TIMESTAMP)
INSERT INTO language
VALUES(NULL,'$name',CURRENT_TIMESTAMP)
INSERT INTO payment
VALUES(NULL,'$customer_id','$staff_id','$rental_id','$amount',CURRENT_TIMESTAMP,CURRENT_TIMESTAMP)
INSERT INTO rental
VALUES(NULL,'$rental_date','$inventory_id','$customer_id','$return_date','$staff_id',CURRENT_TIMESTAMP)
INSERT INTO staff
VALUES(NULL,'$first_name','$last_name','$address_id',
'$email','$store_id','$active','$username','$password',CURRENT_TIMESTAMP)
INSERT INTO store
VALUES(NULL,'$manager_staff_id','$address_id',CURRENT_TIMESTAMP)

Select
SELECT * FROM actor
SELECT * FROM address
SELECT * FROM category
SELECT * FROM city
SELECT * FROM country
SELECT * FROM customer
SELECT * FROM film
SELECT * FROM film_actor
SELECT * FROM inventory
SELECT * FROM language

SELECT * FROM payment
SELECT * FROM rental
SELECT * FROM staff
SELECT * FROM store

Update
UPDATE actor SET first_name='$first_name',last_name='$last_name' WHERE actor_id='$actor_id'
UPDATE address SET address='$address',address2='$address2', district='$district',city_id='$city_id',postal_code='$postal_code',phone='$phone', last_update=CURRENT_TIMESTAMP WHERE address_id='$address_id'
UPDATE category SET name='$name',last_update=CURRENT_TIMESTAMP WHERE category_id='$category_id'
UPDATE city SET city='$city',country_id='$country_id', last_update=CURRENT_TIMESTAMP WHERE city_id='$city_id'
UPDATE country SET country='$country', last_update=CURRENT_TIMESTAMP WHERE country_id='$country_id'
UPDATE customer SET store_id='$store_id',first_name='$first_name',last_name='$last_name',email='$email',address_id='$address_id',active='$active',create_date='$create_date',last_update=CURRENT_TIMESTAMP WHERE customer_id='$customer_id'
UPDATE film SET title='$title',description='$description', release_year='$release_year',language_id='$language_id', original_language_id='$original_language_id',rental_duration='$rental_duration', rental_rate='$rental_rate',length='$length', replacement_cost='$replacement_cost',rating='$rating',special_features='$special_features',last_update=CURRENT_TIMESTAMP,category_id='$category_id' WHERE film_id='$film_id'
UPDATE film_actor SET actor_id='$actor_id',film_id='$film_id', last_update=CURRENT_TIMESTAMP WHERE actor_id=$prevactor_id AND film_id='$prevfilm_id'
UPDATE inventory SET film_id='$film_id',store_id='$store_id',

last_update=CURRENT_TIMESTAMP WHERE inventory_id='$inventory_id'
UPDATE language SET name='$name', last_update=CURRENT_TIMESTAMP WHERE language_id='$language_id'
UPDATE payment SET customer_id='$customer_id',staff_id='$staff_id', rental_id='$rental_id',amount='$amount',payment_date='$payment_date' ,last_update=CURRENT_TIMESTAMP WHERE payment_id='$payment_id'
UPDATE rental SET rental_date='$rental_date',inventory_id='$inventory_id', customer_id='$customer_id',return_date='$return_date',staff_id='$staff_id',last_update=CURRENT_TIMESTAMP WHERE rental_id='$rental_id'
UPDATE staff SET first_name='$first_name',last_name='$last_name', address_id='$address_id',email='$email',store_id='$store_id',active='$active',username='$username',password='$password',last_update=CURRENT_TIMESTAMP WHERE staff_id='$staff_id'
UPDATE store SET manager_staff_id='$manager_staff_id',address_id='$address_id', last_update=CURRENT_TIMESTAMP WHERE store_id='$store_id'

General Statistics of Database

Databases statistics

Database	Collation	Tables	Rows	Data	Indexes	Total	Overhead	Action
hfyyt4_sakilatestfile	latin1_swedish_ci	14	45,273	3.9 MiB	2.4 MiB	6.3 MiB	0 B	Check privileges

Our database is named hfyyt4_sakilatestfile, consists of 14 tables (16 - 2 tables removed due to redundancy), a total of 45,273 rows.

The snapshot above shows a list of tables we have in our database. The rows for each tables are shown as well. 'InnoDB' is used as the database engine as it enables foreign key constraints.

Below are the statistics of each table:

actor

Column	Type	Null	Default
actor_id (Primary)	smallint(5)	No	
first_name	varchar(45)	No	
last_name	varchar(45)	No	
last_update	timestamp	No	CURRENT_TIMESTAMP

category

Column	Type	Null	Default
category_id (Primary)	tinyint(3)	No	
name	varchar(25)	No	
last_update	timestamp	No	CURRENT_TIMESTAMP

city

Column	Type	Null	Default
city_id *(Primary)*	smallint(5)	No	
city	varchar(50)	No	
country_id	smallint(5)	No	
last_update	timestamp	No	CURRENT_TIMESTAMP

country

Column	Type	Null	Default
country_id *(Primary)*	smallint(5)	No	
country	varchar(50)	No	
last_update	timestamp	No	CURRENT_TIMESTAMP

customer

Column	Type	Null	Default
customer_id *(Primary)*	smallint(5)	No	
store_id	tinyint(3)	No	
first_name	varchar(45)	No	
last_name	varchar(45)	No	
email	varchar(50)	Yes	*NULL*
address_id	smallint(5)	No	
active	tinyint(1)	No	1
create_date	datetime	No	
last_update	timestamp	Yes	CURRENT_TIMESTAMP

film

Column	Type	Null	Default
film_id *(Primary)*	smallint(5)	No	
title	varchar(255)	No	
description	text	Yes	*NULL*
release_year	year(4)	Yes	*NULL*
language_id	tinyint(3)	No	
original_language_id	tinyint(3)	Yes	*NULL*
rental_duration	tinyint(3)	No	3
rental_rate	decimal(4,2)	No	4.99
length	smallint(5)	Yes	*NULL*
replacement_cost	decimal(5,2)	No	19.99
rating	enum('G', 'PG', 'PG-13', 'R', 'NC-17')	Yes	G
special_features	set('Trailers', 'Commentaries', 'Deleted Scenes', 'Behind the Scenes')	Yes	*NULL*
last_update	timestamp	No	CURRENT_TIMESTAMP
category_id	tinyint(3)	Yes	*NULL*

film_actor

Column	Type	Null	Default
actor_id *(Primary)*	smallint(5)	No	
film_id *(Primary)*	smallint(5)	No	
last_update	timestamp	No	CURRENT_TIMESTAMP

inventory

Column	Type	Null	Default
inventory_id *(Primary)*	mediumint(8)	No	
film_id	smallint(5)	No	
store_id	tinyint(3)	No	
last_update	timestamp	No	CURRENT_TIMESTAMP

language

Column	Type	Null	Default
language_id *(Primary)*	tinyint(3)	No	
name	char(20)	No	
last_update	timestamp	No	CURRENT_TIMESTAMP

payment

Column	Type	Null	Default
payment_id *(Primary)*	smallint(5)	No	
customer_id	smallint(5)	No	
staff_id	tinyint(3)	No	
rental_id	int(11)	Yes	*NULL*
amount	decimal(5,2)	No	
payment_date	datetime	No	
last_update	timestamp	Yes	CURRENT_TIMESTAMP

rental

Column	Type	Null	Default
rental_id *(Primary)*	int(11)	No	
rental_date	datetime	No	
inventory_id	mediumint(8)	No	
customer_id	smallint(5)	No	
return_date	datetime	Yes	*NULL*
staff_id	tinyint(3)	No	
last_update	timestamp	No	CURRENT_TIMESTAMP

staff

Column	Type	Null	Default
staff_id *(Primary)*	tinyint(3)	No	
first_name	varchar(45)	No	
last_name	varchar(45)	No	
address_id	smallint(5)	No	
email	varchar(50)	Yes	*NULL*
store_id	tinyint(3)	No	
active	tinyint(1)	No	1
username	varchar(16)	No	
password	varchar(40)	Yes	*NULL*
last_update	timestamp	No	CURRENT_TIMESTAMP

staff

store

Column	Type	Null	Default
store_id *(Primary)*	tinyint(3)	No	
manager_staff_id	tinyint(3)	No	
address_id	smallint(5)	No	
last_update	timestamp	No	CURRENT_TIMESTAMP