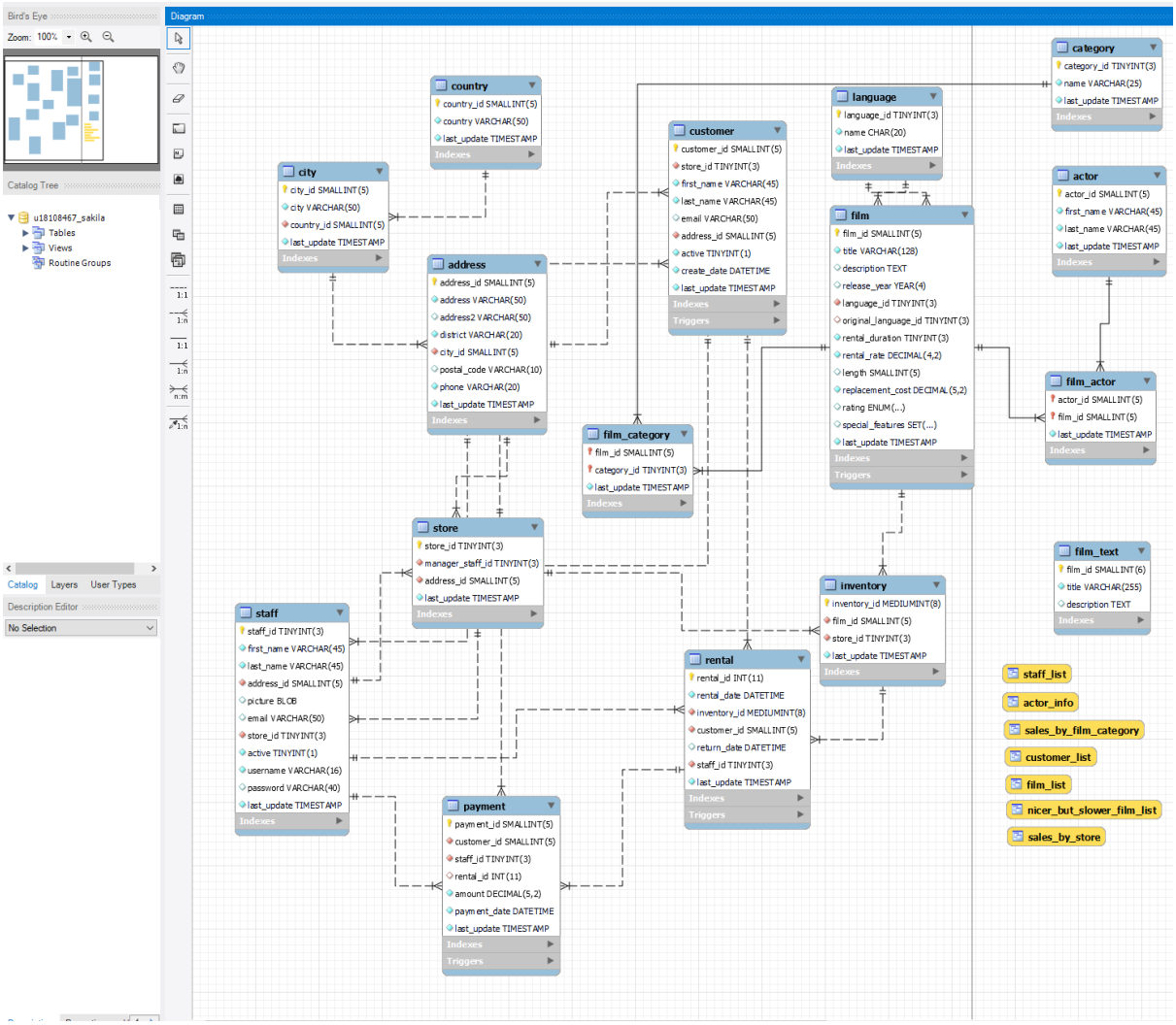
PRACTICAL ASSIGNMENT 4

By Dylan Kapnias (u18108467)

Task 2

2.1



2.2

The ‘store’ relation 4 fields, namely:- ‘store_id’; ‘manager_staff_id’; address_id’; ‘last_update’.

- store_id: This is the primary key of the relation. This is a number that will be used to reference the store. The data type used to store the value in this field is “tinyint(3)”, it is also unsigned, auto incrementing and NOT NULL. This means that the data would occupy 1-byte and be padded to always display 3 characters (as the field does not have the ZERO FILL option, it will be padded with blank spaces), due to it being unsigned it will be able to store the values [0, 255], the auto increment would enable the id to increment if the data inserted into the relation either has the ‘store_id’ field as empty or NULL, and the NOT NULL ensures that the primary key constraint will not be violated.
- manager_staff_id: This is a number that will be used to reference the manager of the store. The data type used to store the value in this field is “tinyint(3)”, it is also unsigned and NOT NULL. This means that the data would occupy 1-byte and be padded to always display 3

characters (as the field does not have the ZERO FILL option, it will be padded with blank spaces), due to it being unsigned it will be able to store the values [0, 255] and the NOT NULL will ensure that the data inserted will not be a NULL value.

- address_id: This is a number that will be used to reference the address of the store. The data type used to store the value in this field is “smallint(5)”, it is also unsigned and NOT NULL. This means that the data would occupy 2-bytes and be padded to always display 5 characters (as the field does not have the ZERO FILL option, it will be padded with blank spaces), due to it being unsigned it will be able to store the values [0, 65535] and the NOT NULL will ensure that the data inserted will not be a NULL value.
- last_update: This is a timestamp that will be used to show the date and time of when the data in a specific record was last updated. The data type used to store the value in this field is “timestamp”, it is also NOT NULL, has a default value of “CURRENT_TIMESTAMP” and has an update rule of “CURRENT_TIMESTAMP”. This means that the values inserted into this field cannot be NULL, when data is inserted into the relation this field will automatically insert the current date and time in the form of a timestamp as the value and when a record in the relation is updated, that specific record’s ‘last_update’ will get its value change to the current date and time in the form of a timestamp.

The ‘store’ relation has 1 PRIMARY KEY, 1 UNIQUE KEY and 2 FOREIGN KEYs’.

The PRIMARY KEY is ‘store_id’.

The UNIQUE KEY is ‘manager_staff_id’ and has an index of ‘idx_unique_manager’.

The first foreign key is ‘address_id’, which references ‘address_id’ from the ‘address’ relation, and has an index of ‘idx_fk_address_id’. This key also has a constraint ‘fk_store_address’ which is ON UPDATE CASCADE, this ensures that when the data in the parent relation is updated, it is also changed in the child relation i.e., when the ‘address_id’ in the ‘address’ relation changes, then it will propagate the change to the ‘address_id’ in the ‘store’ relation.

The second foreign key is ‘manager_staff_id’ which references ‘staff_id’ from the ‘staff’ relation. This key also has a constraint ‘fk_store_staff’ which is ON UPDATE CASCADE, this ensures that when the data in the parent relation is updated, it is also changed in the child relation i.e., when the ‘staff_id’ in the ‘staff’ relation changes, then it will propagate the change to the ‘manager_staff_id’ in the ‘store’ relation.

Task 5

5.1

Github repo:- https://github.com/dylankapnias/COS221_Prac4

5.3

```
String q = "SELECT v.store, e.Genre, e.Stock FROM (" +
                "SELECT * FROM staff JOIN (" +
                    "SELECT i.store_id as sID, COUNT(i.inventory_id) AS Stock, v.category AS Genre FROM " +
                    "inventory AS i JOIN film_list as v ON i.film_id = v.FID GROUP BY v.category ORDER BY " +
                    "i.store_id) " +
                "AS s ON staff.store_id = s.sID) " +
            "AS e JOIN sales_by_store AS v ON v.manager LIKE CONCAT(e.first_name, ' ', e.last_name);";
```