**Databases Laboratory 1: Basic SQL Statements**

Vinicius Beuther, [vinbeu25@student.hh.se](mailto:vinbeu25@student.hh.se), DS4001-2025

Lucca Heineck, [luccou25@student.hh.se](mailto:luccou25@student.hh.se), DS4001-2025

Jan. 2025

# **1 Introduction**

The objective of this lab is to practice your skills in SQL queries. You will be using Data Definition Language (DDL) and Data Manipulation Language (DML) for

* creating a database with multiple tables, and inserting a few entries, according to the given information;
* querying data from several different Databases.

In addition, you will explore a real-world dataset on movies in task 3, using SQL queries

# **2 Task 1**

## **2.1 Task 1.a**

Q1. Create a database with three tables, i.e. Student (according to Table 1), Course (Table 2), and Enrolled (Table 3);

# Lab 1 - Create and Insert in the tables

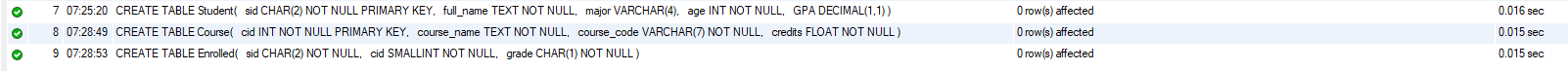
CREATE DATABASE HH\_DB\_Lab1;

USE HH\_DB\_Lab1;

**CREATE TABLE** Student(sid CHAR(2) NOT NULL, full\_name TEXT NOT NULL, major VARCHAR(4), age INT NOT NULL, GPA DECIMAL(2,1), PRIMARY KEY(sid));

**CREATE TABLE** Course(cid INT NOT NULL, course\_name TEXT NOT NULL, course\_code VARCHAR(7) NOT NULL, credits FLOAT NOT NULL, PRIMARY KEY(cid));

**CREATE TABLE** Enrolled(sid CHAR(2) NOT NULL, cid INT NOT NULL, grade CHAR(1) NOT NULL, FOREIGN KEY (sid) REFERENCES Student(sid), FOREIGN KEY (cid) REFERENCES Course(cid));



## **2.2 Task 1.b**

Q1. Select all students above the age of 20;

SELECT \* FROM Student WHERE age > 20;

A screenshot of a computer

Description automatically generated

Q2. Who is the oldest student?

SELECT \* FROM Student WHERE age = (SELECT MAX(age) FROM Student);

A screenshot of a computer

Description automatically generated

Q3. Count the number of students with age below 20;

SELECT COUNT(age) as 'Number of Students with Age < 20' FROM Student WHERE age < 20;

A close up of a text

Description automatically generated

Q4. How many types of majors were these students admitted to?

SELECT COUNT(DISTINCT major) FROM Student;

A black screen with white text

Description automatically generated

Q5. What is the average GPA of students with age above 20?

SELECT AVG(GPA) FROM Student WHERE age > 20;

A black and white screen with white text

Description automatically generated

Q6. What is the average GPA of students studying the Physics major?

SELECT AVG(GPA) FROM Student WHERE major = ‘PHY’;

A black and white screen with white text

Description automatically generated

Q7. What is the average age of students who took Linear algebra courses?

SELECT AVG(age) FROM Student WHERE sid IN (SELECT sid FROM Enrolled WHERE cid = (SELECT cid FROM Course WHERE course\_name = 'Linear algebra'));

A black and white screen with white text

Description automatically generated

Q8. How many courses has Alice registered for?

SELECT COUNT(\*) as 'Courses Registered' FROM Enrolled WHERE sid = (SELECT sid FROM Student WHERE full\_name='Alice

');

A black screen with white text

Description automatically generated

Q9. How many credits has Alice registered?

SELECT SUM(credits) FROM Course WHERE cid IN (SELECT cid FROM Enrolled WHERE sid = (SELECT sid FROM Student WHERE

full\_name='Alice'));

A black screen with white text

Description automatically generated

Q10. How many credits have students with ages below 20 registered to?

SELECT SUM(credits) as 'Credits Registered for Students < 20' FROM Course WHERE cid IN (SELECT cid FROM Enrolled WHERE sid IN (SELECT sid FROM Student WHERE age < 20));

A black screen with white text

Description automatically generated

## **2.2 Task 1.c**

After solving queries above, can you propose two or more queries of practical usage?

Q1. Which courses has Kayle registered for?

SELECT course\_name FROM Course WHERE cid IN (SELECT cid FROM Enrolled WHERE sid = (SELECT sid FROM Student WHERE full\_name='Kayle'));

A black screen with white text

Description automatically generated

Q2.Who is the oldest student?

SELECT full\_name as 'Oldest Student is', age FROM Student WHERE age = (SELECT MAX(age) FROM Student);

A black screen with white text

Description automatically generated

## **3 Task 2**

**3.1 Task 2a**

Solve the following queries:

Q1. What is the most valuable asset in the inventory?

SELECT \* FROM products p ORDER BY unit\_price DESC;

A screenshot of a computer

Description automatically generated

Q2. How much does the entire inventory worth?

SELECT SUM(unit\_price \* quantity\_in\_stock) AS total\_price FROM products p;



Q3. Where is the largest office (in terms of numbers of employees) located?

SELECT o.address, o.city, o.state, count(e.employee\_id) AS employees

FROM offices o, employees e

WHERE o.office\_id = e.office\_id

GROUP BY o.office\_id

ORDER BY employees desc;

A screenshot of a computer

Description automatically generated

Q4. Who sits alone?

SELECT e.first\_name, e.last\_name, e.job\_title, o.address, o.city, o.state

FROM employees e, offices o

WHERE e.office\_id = o.office\_id

AND o.office\_id IN (select office\_id

                 FROM employees e

                 GROUP BY office\_id

                 HAVING COUNT(employee\_id) = 1);



Q5. What is the most common payment method?

SELECT pm.name, COUNT(\*) AS amount

FROM payments p, payment\_methods pm

WHERE p.payment\_method = pm.payment\_method\_id

GROUP BY p.payment\_method

ORDER BY amount DESC;

A screenshot of a computer

AI-generated content may be incorrect.

Q6. Which client seems to be the most important one? Motivate your approach and answer

SELECT client\_id, count(\*) AS invoices\_no

FROM invoices i

GROUP BY client\_id;

A screenshot of a computer

AI-generated content may be incorrect.

Q7. How much do order 2 worth?

SELECT SUM(unit\_price)

FROM orders o, order\_items oi

WHERE o.order\_id = oi.order\_id

AND o.order\_id = 2

ORDER BY unit\_price DESC;

A screen shot of a computer

AI-generated content may be incorrect.

Q8. Which customer has their order delivered?

SELECT c.first\_name, c.last\_name, os.name

FROM orders o, customers c, order\_statuses os

WHERE o.customer\_id = c.customer\_id

AND os.order\_status\_id = o.status

AND o.status = 2;

A screenshot of a computer

AI-generated content may be incorrect.

**3.2 Task 2b**

Q1. (sql\_inventory) How many products cost more than 50?

SELECT \*

FROM products p

WHERE quantity\_in\_stock > 50;

**A screenshot of a computer

AI-generated content may be incorrect.**

Q2. (sql\_inventory) How many products are there in total in stock?

SELECT SUM(quantity\_in\_stock) FROM products p;

A grey rectangular object with white text

AI-generated content may be incorrect.

Q3. (sql\_hr) Information about employees and their addresses.

SELECT e.first\_name, e.last\_name, e.job\_title, o.city AS office\_city, o.address AS office\_address

FROM employees e

JOIN offices o ON e.office\_id = o.office\_id

ORDER BY o.city, e.last\_name, e.first\_name;

A screenshot of a computer

AI-generated content may be incorrect.

Q4. (sql\_hr) What is the highest salary in each office?

SELECT e.first\_name, e.last\_name, e.job\_title, o.city AS office\_city, o.address AS office\_address

FROM employees e

JOIN offices o ON e.office\_id = o.office\_id

ORDER BY o.city, e.last\_name, e.first\_name;

A screenshot of a computer

AI-generated content may be incorrect.

Q5. (sql\_invoicing) What clients have unpaid invoices?

SELECT distinct c.client\_id, c.name, c.address, c.city, c.state, c.phone

FROM clients c

JOIN invoices i ON c.client\_id = i.client\_id

WHERE i.invoice\_total > i.payment\_total;

A screen shot of a address

AI-generated content may be incorrect.

Q6. (sql\_invoicing) What is the total amount paid by each payment method?

SELECT pm.name as payment\_method, SUM(p.amount) as total\_amount

FROM payments p

JOIN payment\_methods pm on p.payment\_method = pm.payment\_method\_id

GROUP BY pm.name

ORDER BY total\_amount desc;

A screenshot of a computer

AI-generated content may be incorrect.

Q7. (sql\_store) What products have been ordered less than 50 times?

SELECT p.product\_id,

    p.name AS product\_name,

    SUM(oi.quantity) as total\_ordered\_quantity

FROM products p

LEFT JOIN order\_items oi on p.product\_id = oi.product\_id

GROUP BY p.product\_id, p.name

HAVING total\_ordered\_quantity < 50 or total\_ordered\_quantity is null

ORDER BY total\_ordered\_quantity asc;

A screenshot of a computer

AI-generated content may be incorrect.

Q8. (sql\_store) Customers details and the total amount for each order.

select o.order\_id,

    CONCAT(c.first\_name, ' ', c.last\_name) as customer\_name,

    o.order\_date,

    SUM(oi.quantity \* oi.unit\_price) as total\_amount

from orders o

join customers c on o.customer\_id = c.customer\_id

JOIN order\_items oi on o.order\_id = oi.order\_id

group by o.order\_id, customer\_name, o.order\_date

order by o.order\_date desc;

A screenshot of a computer

AI-generated content may be incorrect.

## **4 Task 3**

Download the Movie Database shared on the blackboard.

**4.1 Task 3a**

Q1. How many movies have the highest rating?

SELECT COUNT(rating) FROM Ratings WHERE rating = (SELECT MAX(rating) FROM Ratings);



Q2. What are the most common genres in this database?

SELECT genres, count(title\_id) as n FROM titles GROUP BY genres ORDER BY n DESC limit 1;



Q3. Which movie is the longest?

SELECT primary\_title, MAX(runtime\_minutes) FROM Titles;



**4.2 Task 3b**

In addition to the queries above, propose two or more queries of your interesting.

Q1. Which of the series has the biggest number of seasons?

SELECT t.title\_id, t.primary\_title, e.season\_number FROM Titles as t

JOIN Episodes as e

ON t.title\_id = e.episode\_title\_id

WHERE e.season\_number = (SELECT MAX(season\_number) FROM Episodes);



Q2. What is the movie where the oldest actor acted? Also, show his age and name.

SELECT t.primary\_title, p.name, MAX(p.died - p.born) FROM people p, titles t, crew c WHERE c.title\_id = t.title\_id and p.person\_id = c.person\_id;

****

# **5 Conclusion**

Add some reflections/conclusions about the lab, focusing on at least three points listed in the Introduction.

R. In our opinion, this lab allowed us to be familiar with the interface for some of the most used database management systems (MySQL and SQLite3). During this lab we practiced different types of DML and DDL commands to create the requested database, defining different data types for each column and table.

In task 2, we had to handle queries to search, find and cross reference data to get the requested information, learning how to manage and work with foreign keys and some types of JOINs.

In task 3, we faced a real world database containing millions of records among different tables, it allowed us to work with a existent database which was modeled by another people, forcing us to understand the table’s structure and how the database was structed, which types of columns, data and tables, looking for foreign and primary keys when needed, with this lab we learned a lot about important concepts of database management systems and different types of queries.