

Databases Laboratory 1: Basic SQL Statements

Vinicius Beuther, vinbeu25@student.hh.se, DS4001-2025

Lucca Heineck, 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));
```

7 07:25:20	CREATE TABLE Student(sid CHAR(2) NOT NULL PRIMARY KEY, full_name TEXT NOT NULL, major VARCHAR(4), age INT NOT NULL, GPA DECIMAL(1,1))	0 rows(s) affected	0.016
8 07:28:49	CREATE TABLE Course(cid INT NOT NULL PRIMARY KEY, course_name TEXT NOT NULL, course_code VARCHAR(7) NOT NULL, credits FLOAT NOT NULL)	0 rows(s) affected	0.015
9 07:28:53	CREATE TABLE Enrolled(sid CHAR(2) NOT NULL, cid SMALLINT NOT NULL, grade CHAR(1) NOT NULL)	0 rows(s) affected	0.015

2.2 Task 1.b

Q1. Select all students above the age of 20;

```
SELECT * FROM Student WHERE age > 20;
```

sid	full_name	major	age	GPA
c1	Alice	CS	21	4.0
p2	Albert	PHY	22	3.9
NULL	NULL	NULL	NULL	NULL

Q2. Who is the oldest student?

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

sid	full_name	major	age	GPA
p2	Albert	PHY	22	3.9
NULL	NULL	NULL	NULL	NULL

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;
```

	Number of Students with Age < 20
▶ 2	

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

```
SELECT COUNT(DISTINCT major) FROM Student;
```

+	-----+
	COUNT(DISTINCT major)
+	-----+
	4
+	-----+

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

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

+	-----+
	AVG(GPA)
+	-----+
	3.95000
+	-----+

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

```
SELECT AVG(GPA) FROM Student WHERE major = 'PHY';
```

AVG(GPA)
3.80000

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'));
```

AVG(age)
20.0000

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');
```

Courses Registered
3

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'));
```

Credits Registered By Alice
14.5

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));
```

```
+-----+
| Sum of Credits Registered for Students < 20 |
+-----+
|                                     20.5 |
+-----+
```

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'));
```

```
+-----+
| course_name |
+-----+
| Linear algebra |
| Algorithms   |
+-----+
```

Q2. Who is the oldest student?

```
SELECT full_name as 'Oldest Student is', age FROM Student WHERE age = (SELECT MAX(age) FROM Student);
```

```
+-----+-----+
| Oldest Student is | age |
+-----+-----+
| Albert            | 22  |
+-----+-----+
```

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;
```

123 product_id	A-Z name	123 quantity_in_stock	123 unit_price
2	Pork - Bacon,back Peameal	49	4.65
4	Brocolinni - Gaylan, Chinese	90	4.53
3	Lettuce - Romaine, Heart	38	3.35
7	Sweet Pea Sprouts	98	3.29
6	Petit Baguette	14	2.39
9	Longan	67	2.26
5	Sauce - Ranch Dressing	94	1.63
1	Foam Dinner Plate	70	1.21
10	Broom - Push	6	1.09
8	Island Oasis - Raspberry	26	0.74

Q2. How much does the entire inventory worth?

```
SELECT SUM(unit_price * quantity_in_stock) AS total_price FROM products p;
```

123 total_price
1,533.85

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-Z address	A-Z city	A-Z state	123 employees
1	03 Reinke Trail	Cincinnati	OH	4
2	5507 Becker Terrace	New York City	NY	4
3	54 Northland Court	Richmond	VA	4
4	08 South Crossing	Cincinnati	OH	4
5	553 Maple Drive	Minneapolis	MN	3
6	4 Bluestem Parkway	Savannah	GA	1

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);
```

	A-Z first_name	A-Z last_name	A-Z job_title	A-Z address	A-Z city	A-Z state
1	Yvonnnda	Magrannell	Executive Secretary	4 Bluestem Parkway	Savannah	GA

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-Z name	123 amount
1	Credit Card	7
2	Cash	1

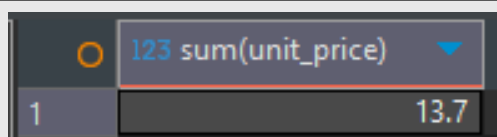
Q6. Which client seems to be the most important one? Motivate your approach and answer

```
SELECT client_id, count(*) AS invoices_no
FROM invoices i
GROUP BY client_id;
```

	123 client_id	123 invoices_no
1	1	5
2	2	1
3	3	5
4	5	6

Q7. How much do order 2 worth?

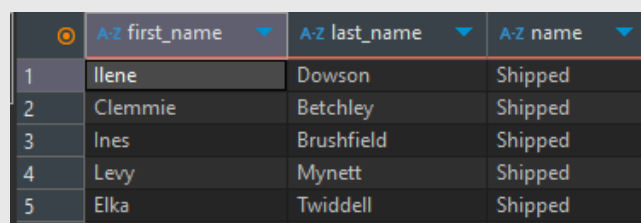
```
SELECT SUM(unit_price)
FROM orders o, order_items oi
WHERE o.order_id = oi.order_id
AND o.order_id = 2
ORDER BY unit_price DESC;
```



	123 sum(unit_price)
1	13.7

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-Z first_name	A-Z last_name	A-Z name
1	Ilene	Dowson	Shipped
2	Clemmie	Betchley	Shipped
3	Ines	Brushfield	Shipped
4	Levy	Mynett	Shipped
5	Elka	Twiddell	Shipped

3.2 Task 2b

Q1. (sql_inventory) How many products cost more than 50?

```
SELECT *
FROM products p
WHERE quantity_in_stock > 50;
```

	123 product_id	A-Z name	123 quantity_in_stock	123 unit_price
1	1	Foam Dinner Plate	70	1.21
2	4	Brocolinni - Gaylan, Chinese	90	4.53
3	5	Sauce - Ranch Dressing	94	1.63
4	7	Sweet Pea Sprouts	98	3.29
5	9	Longan	67	2.26

Q2. (sql_inventory) How many products are there in total in stock?

```
SELECT SUM(quantity_in_stock) FROM products p;
```

	123 sum(quantity_in_stock)
1	552

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-Z first_name	A-Z last_name	A-Z job_title	A-Z office_city	A-Z office_address
1	Keriann	Alloisi	VP Marketing	Cincinnati	03 Reinke Trail
2	Lynde	Aronson	Junior Executive	Cincinnati	08 South Crossing
3	Mindy	Crissil	Staff Scientist	Cincinnati	03 Reinke Trail
4	Cole	Kesterton	Pharmacist	Cincinnati	08 South Crossing
5	Sayer	Matterson	Statistician III	Cincinnati	03 Reinke Trail
6	D'arcy	Nortunen	Account Executive	Cincinnati	03 Reinke Trail
7	Mildrid	Sokale	Geologist II	Cincinnati	08 South Crossing
8	Hazel	Tarbert	General Manager	Cincinnati	08 South Crossing
9	Theresa	Binney	Food Chemist	Minneapolis	553 Maple Drive
10	Estrellita	Daleman	Staff Accountant IV	Minneapolis	553 Maple Drive
11	Ivy	Fearey	Structural Engineer	Minneapolis	553 Maple Drive
12	North	de Clerc	VP Product Manage	New York City	5507 Becker Terrace
13	Elladine	Rising	Social Worker	New York City	5507 Becker Terrace
14	Alaster	Scutchin	Assistant Professor	New York City	5507 Becker Terrace
15	Nisse	Voysey	Financial Advisor	New York City	5507 Becker Terrace
16	Virge	Goodrum	Information System	Richmond	54 Northland Court
17	Kass	Hefferan	Computer Systems	Richmond	54 Northland Court
18	Guthrey	Iacopetti	Office Assistant I	Richmond	54 Northland Court
19	Mirilla	Janowski	Cost Accountant	Richmond	54 Northland Court
20	Yovonnda	Magrannell	Executive Secretary	Savannah	4 Bluestem Parkway

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-Z office_city	A-Z office_address	123 highest_salary
1	Richmond	54 Northland Court	119,241
2	New York City	5507 Becker Terrace	114,257
3	Cincinnati	03 Reinke Trail	110,150
4	Cincinnati	08 South Crossing	93,760
5	Minneapolis	553 Maple Drive	92,710
6	Savannah	4 Bluestem Parkway	63,996

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;
```

	123 client_id	A-Z name	A-Z address	A-Z city	A-Z state	A-Z phone
1	2	Myworks	34267 Glendale Parkway	Huntington	WV	304-659-1170
2	5	Topiclounge	0863 Farmco Road	Portland	OR	971-888-9129
3	3	Yadel	096 Pawling Parkway	San Francisco	CA	415-144-6037
4	1	Vinte	3 Nevada Parkway	Syracuse	NY	315-252-7305

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-2 payment_method	123 total_amount
1	Credit Card	351.38
2	Cash	10

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;
```

	123 product_id	A-2 product_name	123 total_ordered_quantity
1	7	Sweet Pea Sprouts	[NULL]
2	8	Island Oasis - Raspberry	2
3	5	Sauce - Ranch Dressing	3
4	2	Pork - Bacon,back Peameal	7
5	6	Petit Baguette	7
6	10	Broom - Push	7
7	4	Brocolinni - Gaylan, Chinese	8
8	9	Longan	9
9	1	Foam Dinner Plate	16
10	3	Lettuce - Romaine, Heart	28

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;
```

	123 order_id	A-Z customer_name	🕒 order_date	123 total_amount
1	1	Elka Twiddell	2019-01-30	14.96
2	6	Levy Mynett	2018-11-18	81.01
3	7	Ines Brushfield	2018-09-22	64.19
4	2	Ilene Dowson	2018-08-02	30.72
5	8	Clemmie Betchley	2018-06-08	31.06
6	10	Elka Twiddell	2018-04-22	98.62
7	3	Thacher Naseby	2017-12-01	91.2
8	5	Clemmie Betchley	2017-08-25	29.67
9	9	Levy Mynett	2017-07-05	36.4
10	4	Ines Brushfield	2017-01-22	93.73

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);
```

```
sqlite> SELECT COUNT(rating) FROM Ratings WHERE rating = (SELECT MAX(rating) FROM Ratings);
653
```

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;
```

```
Drama|155740
```

Q3. Which movie is the longest?

```
SELECT primary_title, MAX(runtime_minutes) FROM Titles;
```

```
Logistics|51420
```

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);
```

```
tt10903194|Tinder Management|2020
```

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;
```

```
sqlite> 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;
The Vanishing Image|Gustave Field|116
```

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 structured, 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.