

END OF MODULE PRESENTATION

1.0 PART 1: INTRODUCTION

In response to the **high infant and child mortality rate caused by short illnesses in Dille Village, Borno State**, the *Hope Alive Foundation*, an international NGO, established a children's hospital known as **Hope Alive Clinic**.

The foundation initially planned to offer **completely free healthcare**, but to ensure sustainability, patients now pay only **20% of consultation fees**, while all prescribed drugs remain **completely free**.

However, the hospital faced serious **record management challenges**. Patient records, doctor appointments, and prescriptions were stored manually on paper. As a result, records were often lost, prescriptions mismatched, and appointments double-booked. This inefficiency raised serious concerns about the quality of service delivery.

The hospital director, **Mr. Yusuf**, highlighted the urgent need for an **electronic system** to store, update, and retrieve records reliably. I have been tasked with designing and implementing the **database system** that will serve as the backbone of this solution.

Patient Journey Example (Case Scenario)

To better understand the hospital workflow, Mr. Yusuf walked me through a **typical patient's journey**:

- Registration:**

If *Maryam*, a young child suffering from malaria, visits the clinic with her father (*Mr. Hassan, her guardian*), the hospital records both **Maryam's personal details** and **her father's details** as guardian information.

Guardian Reusability:

Later, if Maryam's brother *Zubair* also falls sick and comes with their father, the hospital will again record *Mr. Hassan's details* as guardian for Zubair. This shows how one guardian can be linked to **multiple patients**.

Appointment

After registration, Maryam books an **appointment with a doctor**.

Consultation & Prescription:

Maryam meets the doctor, who examines her and prescribes medication. Depending on her condition, the prescription may include **no drugs** or **multiple drugs**.

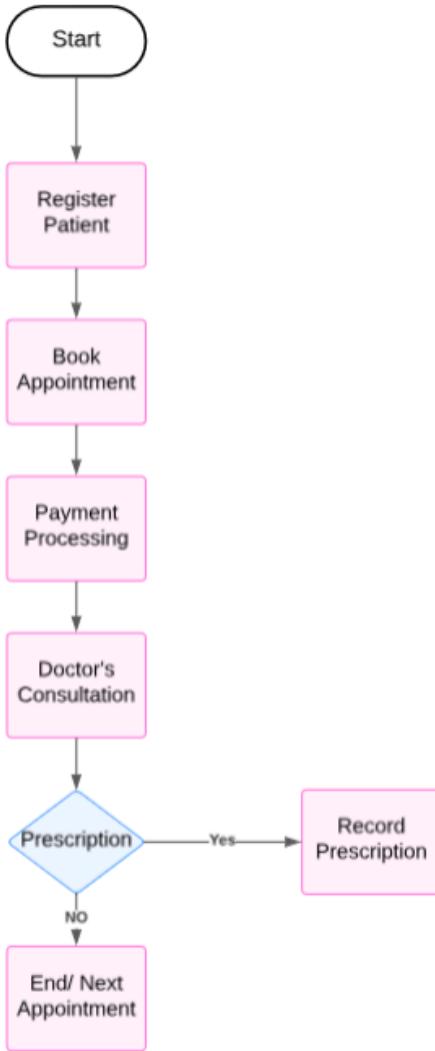
Repeat Visits:

If Maryam later falls sick again, her details are already stored in the hospital's database. Instead of re-entering her information, staff simply create a **new appointment record** linked to her **patient ID**. This ensures that all her appointments and prescriptions are tracked historically, giving doctors access to her complete **medical history** for better diagnosis and treatment.

Doctors & Departments:

Every doctor at Hope Alive Clinic must belong to one of the **10 hospital departments**, and each doctor's **specialization must match their assigned department**.

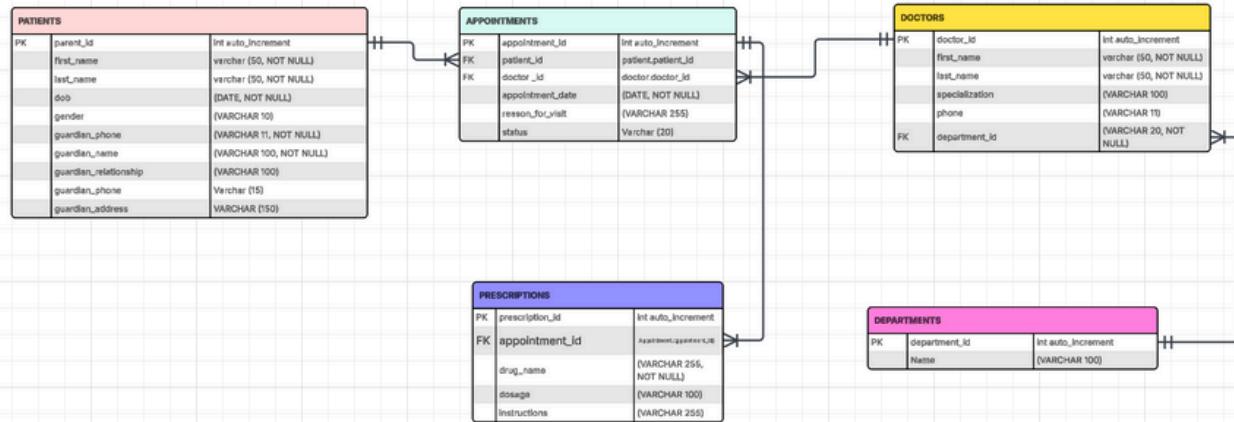
1.1 Hospital Process Workflow Flowchart



1.2 Data Storage & Relationships

The **Hope Alive Children Clinics Database** manages patient care, appointments, and medical records through five entities: **Patients**, **Appointments**, **Doctors**, **Departments**, and **Prescriptions**. These entities work together to track patient details, doctor assignments, departmental roles, and prescriptions for efficient healthcare management.

HOPE ALIVE CHILDREN CLINICS DATABASE



1.3 Database Constraints & Rules

Primary Keys: Every table (patient_id, appointment_id, doctor_id, department_id, prescription_id) has a unique identifier.

Foreign Keys:

patient_id in Appointments → references Patients.

doctor_id in Appointments → references Doctors.

department_id in Doctors → references Departments.

appointment_id in Prescriptions → references Appointments.

NOT NULL: Critical fields like patient details, guardian phone, appointment date, doctor assignment, department name, and drug name must always be entered.

Data Types & Length Rules: Controlled formats for phone numbers, emails, and text fields (e.g., VARCHAR(11) for phone).

Check/Enum Rules: Appointment status restricted to values like Scheduled, Completed, etc.

Trigger Rule: Patient age is auto-calculated from date of birth using a trigger, ensuring accuracy.

```
employees HA_CLINICS_DB Janada_EOM* Janada_EOM_Part2
4  -- Create Patients Table
5 • CREATE TABLE patients (
6     patient_id VARCHAR(10) PRIMARY KEY,
7     first_name VARCHAR(50) NOT NULL,
8     last_name VARCHAR(50) NOT NULL,
9     dob DATE NOT NULL,
10    gender VARCHAR(10) NOT NULL CHECK (gender IN ('Male', 'Female')),
11    guardian_name VARCHAR(100) NOT NULL,
12    guardian_relationship VARCHAR(50) NOT NULL,
13    guardian_phone VARCHAR(15) NOT NULL,
14    guardian_address VARCHAR(150)
15 );
16
17 -- Change delimiter to allow trigger body
18 DELIMITER //
19
20 • -- Trigger for INSERT
21 CREATE TRIGGER check_child_age_insert
22 BEFORE INSERT ON patients
23 FOR EACH ROW
24 BEGIN
25     IF TIMESTAMPOFF(YEAR, NEW.dob, CURDATE()) >= 18 THEN
26         SIGNAL SQLSTATE '45000'
27         SET MESSAGE TEXT = 'Patient must be under 18 years old (INSERT blocked).';

```

Output:

Action Output

```
employees HA_CLINICS_DB Janada_EOM* Janada_EOM_Part2
58 department_id VARCHAR(10) NOT NULL,
59 FOREIGN KEY (department_id) REFERENCES departments(department_id)
60     ON DELETE RESTRICT ON UPDATE CASCADE
61 );
62
63 • CREATE TABLE appointments (
64     appointment_id VARCHAR(10) PRIMARY KEY,
65     patient_id VARCHAR(10) NOT NULL,
66     doctor_id VARCHAR(10) NOT NULL,
67     appointment_date DATE NOT NULL,
68     reason VARCHAR(200),
69     status VARCHAR(20) CHECK (status IN ('Scheduled', 'Completed', 'Cancelled')) NOT NULL,
70     FOREIGN KEY (patient_id) REFERENCES patients(patient_id)
71         ON DELETE CASCADE ON UPDATE CASCADE,
72     FOREIGN KEY (doctor_id) REFERENCES doctors(doctor_id)
73         ON DELETE RESTRICT ON UPDATE CASCADE
74 );
75
76 • CREATE TABLE prescriptions (
77     prescription_id VARCHAR(10) PRIMARY KEY,
78     appointment_id VARCHAR(10) NOT NULL,
79     drug_name VARCHAR(100) NOT NULL,
80     dosage VARCHAR(50) NOT NULL,
81     instructions VARCHAR(200),

```

Output:

Action Output

employees HA_CLINICS_DB Janada_EOM* Janada_EOM_Part2

```

85 • INSERT INTO patients
86 (patient_id, first_name, last_name, dob, gender, guardian_name, guardian_relationship, guardian_phone, guardian_address)
87 VALUES
88 ('P001', 'Hannah', 'Ali', '2025-09-22', 'Female', 'Peter Yohanna', 'Grandfather', '08014443184', 'Brooksstad, Borno State'),
89 ('P002', 'Sarah', 'Lawal', '2023-08-12', 'Female', 'Linda Hassan', 'Uncle', '08024221074', 'North Stevenbury, Borno State'),
90 ('P003', 'Mohammed', 'Saleh', '2023-04-03', 'Male', 'Karen Ali', 'Uncle', '08026198168', 'Phillipschester, Borno State'),
91 ('P004', 'Bala', 'Mohammed', '2025-05-19', 'Male', 'Brandon Gambo', 'Sister', '08089514048', 'Higginshire, Borno State'),
92 ('P005', 'Bala', 'Ibrahim', '2025-05-04', 'Male', 'Tyler Saleh', 'Father', '08058370286', 'Romeromouth, Borno State'),
93 ('P006', 'Maryam', 'Yohanna', '2021-11-16', 'Female', 'Jessica Suleiman', 'Mother', '08094751230', 'Lake Ashley, Borno State'),
94 ('P007', 'Abubakar', 'Gambo', '2022-06-25', 'Male', 'Michael Bala', 'Uncle', '08033782945', 'Port Joel, Borno State'),
95 ('P008', 'Aisha', 'Danladi', '2024-07-10', 'Female', 'Susan Adamu', 'Aunt', '08022154768', 'West Amanda, Borno State'),
96 ('P009', 'Yusuf', 'Hassan', '2023-01-28', 'Male', 'Anthony Ibrahim', 'Father', '08015469733', 'Lake Jasmine, Borno State'),
97 ('P010', 'Zainab', 'Adamu', '2022-02-14', 'Female', 'Emily Danjuma', 'Mother', '08086321574', 'West Lisa, Borno State'),
98 ('P011', 'Ibrahim', 'Abdullahi', '2024-12-19', 'Male', 'Grace Bala', 'Grandmother', '08061735829', 'Port Brian, Borno State'),
99 ('P012', 'Naomi', 'Suleiman', '2025-01-23', 'Female', 'Ahmed Lawal', 'Father', '08094563712', 'East Monica, Borno State'),
100 ('P013', 'Usman', 'Okeke', '2024-09-08', 'Male', 'Hauwa Saleh', 'Mother', '08062354897', 'Lake David, Borno State'),
101 ('P014', 'Esther', 'Danladi', '2021-03-11', 'Female', 'Simon Mohammed', 'Uncle', '08031928754', 'New Kevin, Borno State'),
102 ('P015', 'Yahaya', 'Aliyu', '2025-02-27', 'Male', 'Ruth Hassan', 'Mother', '08051472893', 'Lake Rebecca, Borno State'),
103 ('P016', 'Fatima', 'Gambo', '2023-07-16', 'Female', 'Isa Adamu', 'Father', '08021836479', 'Port Deborah, Borno State'),
104 ('P017', 'David', 'Yohanna', '2021-05-04', 'Male', 'John Bala', 'Brother', '08078621459', 'South Joshua, Borno State'),
105 ('P018', 'Hadiza', 'Mohammed', '2022-08-20', 'Female', 'Maryam Usman', 'Mother', '08096251438', 'North Samuel, Borno State'),
106 ('P019', 'Simon', 'Saleh', '2024-11-29', 'Male', 'Paul Ali', 'Father', '08085261947', 'Lake Daniel, Borno State'),
107 ('P020', 'Rebecca', 'Bala', '2023-10-13', 'Female', 'Deborah Ibrahim', 'Sister', '08074169382', 'West Martha, Borno State'),
108 ('P021', 'Andrew', 'Abdullahi', '2025-03-04', 'Male', 'Martha Suleiman', 'Mother', '08075316289', 'East Lydia, Borno State'),

```

employees HA_CLINICS_DB Janada_EOM* Janada_EOM_Part2

```

265
266 • INSERT INTO appointments (appointment_id, patient_id, doctor_id, appointment_date, reason, status) VALUES
267 ('A001', 'P003', 'DR001', '2022-04-15', 'Fever and cough', 'Completed'),
268 ('A002', 'P010', 'DR022', '2023-06-12', 'Heart murmur check', 'Scheduled'),
269 ('A003', 'P045', 'DR041', '2024-01-18', 'Ear infection', 'Completed'),
270 ('A004', 'P020', 'DR005', '2021-10-29', 'Growth monitoring', 'Completed'),
271 ('A005', 'P099', 'DR033', '2025-02-11', 'Suspected leukemia', 'Scheduled'),
272 ('A006', 'P011', 'DR026', '2022-07-04', 'Seizure evaluation', 'Cancelled'),
273 ('A007', 'P074', 'DR046', '2023-03-09', 'Skin rash', 'Completed'),
274 ('A008', 'P050', 'DR016', '2021-12-21', 'Accidental fall', 'Completed'),
275 ('A009', 'P022', 'DR006', '2022-09-17', 'Newborn jaundice', 'Completed'),
276 ('A010', 'P087', 'DR038', '2024-08-25', 'Sports injury', 'Scheduled'),
277 ('A011', 'P005', 'DR002', '2023-02-14', 'Malaria treatment', 'Completed'),
278 ('A012', 'P033', 'DR031', '2021-08-07', 'Cancer screening', 'Completed'),
279 ('A013', 'P041', 'DR010', '2025-01-20', 'Newborn follow-up', 'Scheduled'),
280 ('A014', 'P072', 'DR021', '2023-07-02', 'Chest pain', 'Completed'),
281 ('A015', 'P014', 'DR048', '2022-11-09', 'Skin infection', 'Cancelled'),
282 ('A016', 'P018', 'DR017', '2024-03-14', 'High fever', 'Completed'),
283 ('A017', 'P008', 'DR005', '2021-09-05', 'Nutritional check', 'Completed'),
284 ('A018', 'P023', 'DR027', '2022-12-18', 'Epilepsy management', 'Scheduled'),
285 ('A019', 'P067', 'DR034', '2023-05-30', 'Genetic counseling', 'Completed'),
286 ('A020', 'P030', 'DR004', '2021-06-22', 'Infectious disease', 'Completed'),
287 ('A021', 'P056', 'DR012', '2025-04-19', 'Heart surgery consult', 'Scheduled'),
288 ('A022', 'P002', 'DR044', '2023-01-12', 'Tonsillitis', 'Completed'),

```

1.4

first i did for all patients with status scheduled, since scheduled means future/upcoming appointments

MySQL Workbench

Local instance MySQL80 × MySQL Model* × EER Diagram ×

File Edit View Query Database Server Tools Scripting Help

Navigator: employees HA_CLINICS_DB* Janada_EOM* Janada_EOM_Part2

SCHEMAS

Filter objects

- employees
- grodt_university
- hope_alive_clinics
- Tables
 - appointments
 - departments
 - doctors
 - patients
 - Columns
 - Indexes

Administration Schemas

Information

Schema: grodt_university

Result Grid | Filter Rows: Export: Wrap Cell Content: Result 1 × Read Only

patient_id	last_name	dob	status
P010	Adamu	2022-02-14	Scheduled
P099	Mohammed	2024-09-28	Scheduled
P087	Danladi	2022-01-21	Scheduled
P041	Adamu	2021-07-30	Scheduled
P023	Lawal	2023-05-15	Scheduled
P056	Lawal	2024-05-29	Scheduled
P044	Hassan	2025-01-05	Scheduled
P012	Suleiman	2025-01-23	Scheduled
P021	Abdullahi	2025-03-04	Scheduled
P073	Danladi	2022-05-18	Scheduled
P085	Yohanna	2025-02-19	Scheduled
P060	Hassan	2022-01-07	Scheduled

Output

Action Output

#	Time	Action	Message	Duration / Fetch
1	13:04:40	drop database femi_hospital_pms_db	7 row(s) affected	0.250 sec
2	14:07:02	select p.patient_id, p.last_name, p.dob, a.status from patients p join appointments a on p.patient_id = a.patient_id where a.status = 'Scheduled'	Error Code: 1146. Table 'employees.patients' doesn't exist	0.000 sec
3	14:07:18	select p.patient_id, p.last_name, p.dob, a.status from patients p join appointments a on p.patient_id = a.patient_id where a.status = 'Scheduled'	45 row(s) returned	0.000 sec / 0.000 sec

Object Info Session

Queried Completed

Then i now picked one patient and their date of birth to check for his next scheduled date, since a previous record might show, i now added appointment date so i can select the most recent one.

MySQL Workbench

Local instance MySQL80 × MySQL Model* × EER Diagram ×

File Edit View Query Database Server Tools Scripting Help

Navigator: employees HA_CLINICS_DB* Janada_EOM* Janada_EOM_Part2

SCHEMAS

Filter objects

- employees
- grodt_university
- hope_alive_clinics
- Tables
 - appointments
 - departments
 - doctors
 - patients
 - Columns
 - Indexes

Administration Schemas

Information

Schema: grodt_university

Result Grid | Filter Rows: Export: Wrap Cell Content: Result 2 × Read Only

patient_id	last_name	dob	status	appointment_date
P043	Danladi	2022-09-26	Scheduled	2025-05-19
P043	Danladi	2022-09-26	Scheduled	2023-11-04

Output

Action Output

#	Time	Action	Message	Duration / Fetch
2	14:07:02	select p.patient_id, p.last_name, p.dob, a.status from patients p join appointments a on p.patient_id = a.patient_id where a.status = 'Scheduled'	Error Code: 1146. Table 'employees.patients' doesn't exist	0.000 sec
3	14:07:18	select p.patient_id, p.last_name, p.dob, a.status from patients p join appointments a on p.patient_id = a.patient_id where a.status = 'Scheduled'	45 row(s) returned	0.000 sec / 0.000 sec
4	14:08:46	select p.patient_id, p.last_name, p.dob, a.status, a.appointment_date from patients p join appointments a on	2 row(s) returned	0.000 sec / 0.000 sec

Object Info Session

Queried Completed

for number 2, i identified the tables i need to join patients and departments, i identified the patient to appointments then doctors and then departments, then i counted a distinct patient id so it doesnt get repeated. then grouped by department name, and got the number of patients that saw doctors in each departments or the number of patients assigned to each department.

MySQL Workbench

Local instance MySQL80 ×

File Edit View Query Database Server Tools Scripting Help

Navigator: employees HA_CLINICS_DB Janada_EOM* ×

SCHEMAS

Tables

hope_olive_clinic

Employees

Prescriptions

Departments

Doctors

Patients

Appointments

Filter objects

Limit to 1000 rows

26 join appointments a on p.patient_id = a.patient_id
 27 join doctors doc on a.doctor_id = doc.doctor_id
 28 join departments d on doc.department_id = d.department_id
 29 group by d.department_name
 30 order by Number_of_Patients;
 31
 32 -- Retrieve all prescriptions given by all doctors in the last 6 months
 33
 34 • select d.first_name as doctor_first_name,
 35 d.last_name as doctor_last_name,

Result Grid | Filter Rows: Export: Wrap Cell Content: 15

Table: prescriptions

Columns:

prescription_id	appointment_id	drug_name	dosage	instructions
varchar(1)	PK	varchar(1)	varchar(1)	varchar(5)
				varchar(2)

Result 15 × Read Only

Output

Action Output

#	Time	Action	Message	Duration / Fetch
82	22:56:57	select p.last_name, p.dob, a.status from patients p join appointments a on p.patient_id = a.patient_id where...	45 row(s) returned	0.000 sec / 0.000 sec
83	22:57:23	select p.last_name, p.dob, a.status, a.appointment_date from patients p join appointments a on p.patient_id ...	2 row(s) returned	0.000 sec / 0.000 sec
84	22:58:42	select d.department_name, count(distinct p.patient_id) as Number_of_Patients from patients p join appointment...	10 row(s) returned	0.000 sec / 0.000 sec

Object Info Session

Query Completed

To Retrieve all prescriptions given by a specific doctor in the last 6 months. first i retrieve prescriptions given by all doctors in the last 6 months then i picked a name

MySQL Workbench

Local instance MySQL80 ×

File Edit View Query Database Server Tools Scripting Help

Navigator: employees HA_CLINICS_DB Janada_EOM* ×

SCHEMAS

Tables

hope_olive_clinic

Employees

Prescriptions

Departments

Doctors

Patients

Appointments

Filter objects

Limit to 1000 rows

36 pr.prescription_id,
 37 pr.d Execute the statement under the keyboard cursor
 38 pr.dosage,
 39 a.appointment_date
 40 from prescriptions pr
 41 join appointments a on pr.appointment_id = a.appointment_id
 42 join doctors d on a.doctor_id = d.doctor_id
 43 where a.appointment_date >= date_sub(curdate(), interval 6 Month)
 44 order by d.last_name, a.appointment_date desc;
 45

Result Grid | Filter Rows: Export: Wrap Cell Content: 16

Table: prescriptions

Columns:

prescription_id	appointment_id	drug_name	dosage	instructions
varchar(1)	PK	varchar(1)	varchar(1)	varchar(5)
				varchar(2)

Result 16 × Read Only

Output

Action Output

#	Time	Action	Message	Duration / Fetch
83	22:57:23	select p.last_name, p.dob, a.status, a.appointment_date from patients p join appointments a on p.patient_id ...	2 row(s) returned	0.000 sec / 0.000 sec
84	22:58:42	select d.department_name, count(distinct p.patient_id) as Number_of_Patients from patients p join appointment...	10 row(s) returned	0.000 sec / 0.000 sec
85	22:59:17	select d.first_name as doctor_first_name, d.last_name as doctor_last_name, pr.prescription_id, pr.drug_name, p...	13 row(s) returned	0.000 sec / 0.000 sec

Object Info Session

Query Completed

Retrieve all prescriptions given by a specific doctor in the last 6 months. doctor Naomi Mohammed

MySQL Workbench

Local instance MySQL80 ×

File Edit View Query Database Server Tools Scripting Help

Navigator: employees HA_CLINICS_DB Janada_EOM* ×

SCHEMAS

Tables

hope_olive_clinics

Employees

Appointments

Departments

Doctors

Patients

Prescriptions

Columns

Indexes

Foreign Keys

Triggers

Administration Schemas

Information

Table: prescriptions

Columns:

prescription_id	appointment_id	drug_name	dosage	instructions
varchar(1)	PK	varchar(1)	varchar(1)	varchar(5)
				varchar(2)

Result Grid | Filter Rows: Export: Wrap Cell Content: Result 19 × Read Only

Output

Action Output

#	Time	Action	Message	Duration / Fetch
86	23:06:09	select d.first_name as doctor_first_name, d.last_name as doctor_last_name, pr.prescription_id, pr.drug_name, p...	1 row(s) returned	0.000 sec / 0.000 sec
87	23:06:27	select d.first_name as doctor_first_name, d.last_name as doctor_last_name, pr.prescription_id, pr.drug_name, p...	13 row(s) returned	0.016 sec / 0.000 sec
88	23:07:17	select d.first_name as doctor_first_name, d.last_name as doctor_last_name, pr.prescription_id, pr.drug_name, p...	2 row(s) returned	0.000 sec / 0.000 sec

Object Info Session

Query Completed

```

51    pr.dosage,
52    a.appointment_date
53    from prescriptions pr
54    join appointments a on pr.appointment_id = a.appointment_id
55    join doctors d on a.doctor_id = d.doctor_id
56    where d.first_name = 'Naomi'
57    and d.last_name = 'Mohammed'
58    and a.appointment_date >= date_sub(curdate(), interval 6 Month)
59    order by a.appointment_date desc;
60

```

Show all patients who have had more than 3 appointments in the last year

MySQL Workbench

Local instance MySQL80 ×

File Edit View Query Database Server Tools Scripting Help

Navigator: employees HA_CLINICS_DB Janada_EOM* ×

SCHEMAS

Tables

hope_olive_clinics

Employees

Appointments

Departments

Doctors

Patients

Prescriptions

Columns

Indexes

Foreign Keys

Triggers

Administration Schemas

Information

Table: prescriptions

Columns:

prescription_id	appointment_id	drug_name	dosage	instructions
varchar(1)	PK	varchar(1)	varchar(1)	varchar(5)
				varchar(2)

Result Grid | Filter Rows: Export: Wrap Cell Content: Result 20 × Read Only

Output

Action Output

#	Time	Action	Message	Duration / Fetch
87	23:06:27	select d.first_name as doctor_first_name, d.last_name as doctor_last_name, pr.prescription_id, pr.drug_name, p...	13 row(s) returned	0.016 sec / 0.000 sec
88	23:07:17	select d.first_name as doctor_first_name, d.last_name as doctor_last_name, pr.prescription_id, pr.drug_name, p...	2 row(s) returned	0.000 sec / 0.000 sec
89	23:12:40	select d.first_name, d.last_name, count(a.appointment_id) as Total_Appointments from appointments a join doc...	3 row(s) returned	0.000 sec / 0.000 sec

Object Info Session

Query Completed

```

60
61    -- Find the top 3 doctors who had the highest number of appointments in the last year.
62    select d.first_name, d.last_name, count(a.appointment_id) as Total_Appointments
63    from appointments a
64    join doctors d on a.doctor_id = d.doctor_id
65    where a.appointment_date >= date_sub(curdate(), interval 1 year)
66    group by d.doctor_id, d.first_name, d.last_name
67    order by Total_Appointments desc
68    limit 3;
69

```

2.0 PART 2

What is the total amount paid in salaries for contracts that ended before December 31, 2000?

MySQL Workbench - Local instance MySQL80 - MySQL Model - EER Diagram

Schema: employees

```

1 • use employees;
2 -- What is the total amount paid in salaries for contracts that ended before December 31,2000?
3
4 • select sum(salary) as Total_salaries
5   from salaries
6  where to_date < '2000-12-31';
7
8 -- How many employees have "Smith" in their last name? (0.5 marks)
9
10 • select count(last_name) as Employee_last_name

```

Result Grid

Total_salaries
46366439205

Result 7

Action Output

#	Time	Action	Message	Duration / Fetch
100	23:50:58	select emp_no, salary, case when salary > 100000 then 'High' when salary between 50000 and 100000 then ...	1000 row(s) returned	0.813 sec / 0.000 sec
101	23:51:33	select count(last_name) as Employee_last_name from employees where last_name like "%Smith%" LIMIT 0, 1000	1 row(s) returned	0.313 sec / 0.000 sec
102	23:52:19	select sum(salary) as Total_salaries from salaries where to_date < '2000-12-31' LIMIT 0, 1000	1 row(s) returned	0.875 sec / 0.000 sec

Object Info Session Query Completed

How many employees have "Smith" in their last name? (0.5 marks)

MySQL Workbench - Local instance MySQL80 - MySQL Model - EER Diagram

No object selected

```

4 • select sum(salary) as Total_salaries
5   from salaries
6  where to_date < '2000-12-31';
7
8 -- How many employees have "Smith" in their last name? (0.5 marks)
9
10 • select count(last_name) as Number_of_smiths
11   from employees
12  where last_name like '%Smith%';
13

```

Result Grid

Number_of_smiths
155

Result 4

Action Output

#	Time	Action	Message	Duration / Fetch
2	06:06:12	select emp_no, salary, case when salary > 100000 then 'High' when salary between 50000 and 100000 then ...	1000 row(s) returned	0.985 sec / 0.000 sec
3	06:08:15	select emp_no, salary, case when salary > 100000 then 'High' when salary between 50000 and 100000 then ...	1000 row(s) returned	0.953 sec / 0.000 sec
4	06:14:35	select count(last_name) as Number_of_smiths from employees where last_name like "%Smith%" LIMIT 0, 1000	1 row(s) returned	0.500 sec / 0.000 sec

Object Info Session

Using the salaries table (2 marks) a. create a column that categorizes employees into salary brackets

So i ordered in ascending order

employees HA_CLINICS_DB Janada_EOM* Janada_EOM_Part2* ×

15 -- a. create a column that categorizes employees into salary brackets
 16 • select emp_no, salary,
 17 case
 18 when salary > 100000 then 'High'
 19 when salary between 50000 and 100000 then 'Medium'
 20 when salary < 50000 then 'Low'
 21 end as Categories_of_earners
 22 from salaries
 23 order by salary asc;
 24

Result Grid Filter Rows: Export: Wrap Cell Content: Fetch rows:

emp_no	salary	Categories_of_earners
49239	38735	Low
15830	38812	Low
64198	38836	Low
50419	38850	Low
34707	38851	Low
49239	38859	Low
12444	38874	Low
52167	38888	Low
95705	38928	Low
48744	39006	Low
15830	39012	Low
14573	39020	Low
65904	39035	Low

Result 2 ×

Output

Action Output

#	Time	Action	Message
1	06:05:54	select emp_no, salary, case when salary > 100000 then 'High' when salary between 50000 and 100000 then 'M... 1000 row(s) returned	
2	06:06:12	select emp_no, salary, case when salary > 100000 then 'High' when salary between 50000 and 100000 then 'M... 1000 row(s) returned	

While in descending order

employees HA_CLINICS_DB Janada_EOM* Janada_EOM_Part2* Limit to 1000 rows

```

14  -- Using the salaries table (2 marks)
15  -- a. Execute the statement under the keyboard cursor.
16  • select emp_no, salary,
17  case
18    when salary > 100000 then 'High'
19      when salary between 50000 and 100000 then 'Medium'
20      when salary < 50000 then 'Low'
21    end as Categories_of_earners
22  from salaries
23  order by salary desc;

```

Result Grid | Filter Rows: Export: Wrap Cell Content: Fetch rows: Result 3 × Read Only

emp_no	salary	Categories_of_earners
43624	158220	High
43624	157821	High
47978	155709	High
109334	155377	High
109334	155190	High
109334	154888	High
109334	154885	High
80823	154459	High
43624	153458	High
43624	153166	High
47978	151929	High
80823	151768	High
109334	151484	High

Action Output

#	Time	Action	Message	Duration / Fetch
1	06:05:54	select emp_no, salary, case when salary > 100000 then 'High' when salary between 50000 and 100000 then 'M... 1000 row(s) returned	1000 row(s) returned	1.390 sec / 0.000 sec
2	06:06:12	select emp_no, salary, case when salary > 100000 then 'High' when salary between 50000 and 100000 then 'M... 1000 row(s) returned	1000 row(s) returned	0.985 sec / 0.000 sec
3	06:08:15	select emp_no, salary, case when salary > 100000 then 'High' when salary between 50000 and 100000 then 'M... 1000 row(s) returned	1000 row(s) returned	0.953 sec / 0.000 sec

Create a column that applies a bonus of 10% to salaries below 60,000 and a 5% bonus to others

added 5% bonus to employees earning above 60,000

MySQL Workbench Local instance MySQL80 ×

File Edit View Query Database Server Tools Scripting Help

Navigator: employees HA_CLINICS_DB Janada_EOM* Janada_EOM_Part2* Limit to 1000 rows

```

22  from salaries
23  order by salary desc;

24

25  -- Create a column that applies a bonus of 10% to salaries below 60,000 and a 5% bonus to others.
26  • select emp_no, salary,
27  case
28    when salary < 60000 then salary * 1.10
29    else salary * 1.05
30  end as christmas_bonus
31  from salaries;

```

Result Grid | Filter Rows: Export: Wrap Cell Content: Fetch rows: Result 5 × Read Only

emp_no	salary	christmas_bonus
10001	60117	63122.85
10001	62102	65207.10
10001	66074	69377.70
10001	66596	69925.80
10001	66961	70309.05
10001	71046	74598.30
10001	74333	78049.65
10001	75286	79050.30
10001	75994	79730.70
10001	76884	80728.20
10001	80013	84013.65
10001	81025	85076.25
10001	81097	85151.85

Action Output

#	Time	Action	Message	Duration / Fetch
8	06:19:43	select first_name, last_name, salaries, emp_no, case when salaries < 60000 then salaries * 1.10 else salaries * ... Error Code: 1054. Unknown column 'first_name' in field list'	Error Code: 1054. Unknown column 'first_name' in field list'	0.000 sec
9	06:20:04	select salaries, emp_no, case when salaries < 60000 then salaries * 1.10 else salaries * 1.05 end as christmas... Error Code: 1054. Unknown column 'salaries' in field list'	Error Code: 1054. Unknown column 'salaries' in field list'	0.016 sec
10	06:21:17	select emp_no, salary, case when salary < 60000 then salary * 1.10 else salary * 1.05 end as christmas_bonus... 1000 row(s) returned	1000 row(s) returned	0.015 sec / 0.000 sec

added 10% bonus to employees earning below 60,000

MySQL Workbench

Local instance MySQL80 ×

File Edit View Query Database Server Tools Scripting Help

Navigator: employees HA_CLINICS_DB Janada_EOM* Janada_EOM_Part2*

SCHEMAS Filter objects

- cooperativesdb
- customers
- employees
 - Tables
 - departments
 - dept_emp
 - dept_manager
 - employees
 - salaries
 - titles
 - Views

Administration Schemas

Information: No object selected

Result Grid | Filter Rows: Export: Wrap Cell Content: Fetch rows:

emp_no	salary	christmas_bonus
49239	38735	42609.50
15830	38812	42693.20
64198	38836	42719.60
50419	38850	42735.00
34707	38851	42736.10
49239	38859	42744.90
12444	38874	42761.40
52167	38888	42776.80
95705	38928	42820.80
48744	39006	42906.60
15830	39012	42913.20
14573	39020	42922.00
65904	39035	42938.50

Result 7 × Read Only

Action Output

#	Time	Action	Message	Duration / Fetch
11	06.24.03	select emp_no, salary, case when salary < 60000 then salary * 1.10 else salary * 1.05 end as christmas_bonus	Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds to your MySQL se...	0.000 sec
12	06.24.20	select emp_no, salary, case when salary < 60000 then salary * 1.10 else salary * 1.05 end as christmas_bonus	1000 row(s) returned	0.937 sec / 0.000 sec
13	06.24.31	select emp_no, salary, case when salary < 60000 then salary * 1.10 else salary * 1.05 end as christmas_bonus	1000 row(s) returned	0.953 sec / 0.000 sec

Object Info Session

Add a column that displays Active for contracts where to_date is later than today's date (current_date) and Expired otherwise.

Expired Contracts

MySQL Workbench

Local instance MySQL80 ×

File Edit View Query Database Server Tools Scripting Help

Navigator: employees HA_CLINICS_DB Janada_EOM* Janada_EOM_Part2*

SCHEMAS Filter objects

- cooperativesdb
- customers
- employees
 - Tables
 - departments
 - dept_emp
 - dept_manager
 - employees
 - salaries
 - titles
 - Views

Administration Schemas

Information: No object selected

Result Grid | Filter Rows: Export: Wrap Cell Content: Fetch rows:

emp_no	salary	from_date	to_date	contract_status
10001	60117	1986-06-26	1987-05-26	Expired
10001	62102	1987-06-26	1988-05-25	Expired
10001	66074	1988-06-25	1989-05-25	Expired
10001	66596	1989-06-25	1990-06-25	Expired
10001	66961	1990-06-25	1991-06-25	Expired
10001	71046	1991-06-25	1992-06-24	Expired
10001	74333	1992-06-24	1993-06-24	Expired
10001	75286	1993-06-24	1994-06-24	Expired
10001	75994	1994-06-24	1995-06-24	Expired
10001	76884	1995-06-24	1996-06-23	Expired
10001	80013	1996-06-23	1997-06-23	Expired
10001	81025	1997-06-23	1998-06-23	Expired
10001	81097	1998-06-23	1999-06-23	Expired

Result 8 × Read Only

Action Output

#	Time	Action	Message	Duration / Fetch
12	06.24.20	select emp_no, salary, case when salary < 60000 then salary * 1.10 else salary * 1.05 end as christmas_bonus	1000 row(s) returned	0.937 sec / 0.000 sec
13	06.24.31	select emp_no, salary, case when salary < 60000 then salary * 1.10 else salary * 1.05 end as christmas_bonus	1000 row(s) returned	0.953 sec / 0.000 sec
14	06.28.17	select emp_no, salary, from_date, to_date, case when to_date > current_date() then 'Active' else 'Expired' end...	1000 row(s) returned	0.000 sec / 0.015 sec

Object Info Session

Active Contracts

MySQL Workbench

Local instance MySQL80 ×

File Edit View Query Database Server Tools Scripting Help

Navigator Schemas Administration Schemas

Information No object selected

Result Grid | Filter Rows: active | Export | Wrap Cell Content: | Fetch rows: |

Schema: employees | Table: Janada_EOM_Part2* | Limit to 1000 rows |

```

35 • select emp_no,
36 salary,
37 from_date,
38 to_date,
39 case
40 when to_date > current_date() then 'Active'
41 else 'Expired'
42 end as contract_status
43 from salaries;

```

emp_no	salary	from_date	to_date	contract_status
10001	88958	2002-06-22	9999-01-01	Active
10002	72527	2001-08-02	9999-01-01	Active
10003	43111	2001-12-01	9999-01-01	Active
10004	74057	2001-11-27	9999-01-01	Active
10005	94692	2001-09-09	9999-01-01	Active
10006	59755	2001-08-02	9999-01-01	Active
10007	88076	2002-02-07	9999-01-01	Active
10009	94409	2002-02-14	9999-01-01	Active
10010	80324	2001-11-23	9999-01-01	Active
10012	54423	2001-12-16	9999-01-01	Active
10013	68901	2001-10-16	9999-01-01	Active
10014	60598	2001-12-27	9999-01-01	Active
10016	77935	2002-02-10	9999-01-01	Active

Result 8 ×

Action Output

Time	Action	Message	Duration / Fetch
12 06:24:20	select emp_no, salary, case when salary < 60000 then salary * 1.10 else salary * 1.05 end as christmas_bonu...	1000 row(s) returned	0.937 sec / 0.000 sec
13 06:24:31	select emp_no, salary, case when salary < 60000 then salary * 1.10 else salary * 1.05 end as christmas_bonu...	1000 row(s) returned	0.953 sec / 0.000 sec
14 06:28:17	select emp_no, salary, from_date, to_date, case when to_date > current_date() then 'Active' else 'Expired' end...	1000 row(s) returned	0.000 sec / 0.015 sec

Object Info Session

Use a subquery with an IN operator inside a WHERE clause to obtain the employee number (emp_no), department number (dept_no), and contract start date (from_date) from the dept_manager table. Retrieve only data about managers born in or after 1955. (2 marks)

MySQL Workbench

Local instance MySQL80 ×

File Edit View Query Database Server Tools Scripting Help

Navigator Schemas Administration Schemas

Information No object selected

Result Grid | Filter Rows: | Export/Import: | Wrap Cell Content: | Fetch rows: |

Schema: employees | Table: Janada_EOM_Part2* | Limit to 1000 rows |

```

44
45 -- filter managers based on birth year 1955 inside a subquery.
46 • select emp_no, dept_no, from_date
47   from dept_manager
48 where emp_no in (
49   select emp_no
50     from employees
51   where birth_date >= '1955-01-01'
52 );
53

```

emp_no	dept_no	from_date
110022	d001	1985-01-01
110039	d001	1991-10-01
110085	d002	1985-01-01
110114	d002	1989-12-17
110228	d003	1992-03-21
110303	d004	1985-01-01
110344	d004	1988-09-09
110420	d004	1996-08-30
110511	d005	1985-01-01
110567	d005	1992-04-25
110725	d006	1985-01-01
110800	d006	1991-09-12
110854	d006	1994-06-28

dept_manager 9 ×

Action Output

Time	Action	Message	Duration / Fetch
13 06:24:31	select emp_no, salary, case when salary < 60000 then salary * 1.10 else salary * 1.05 end as christmas_bonu...	1000 row(s) returned	0.953 sec / 0.000 sec
14 06:28:17	select emp_no, salary, from_date, to_date, case when to_date > current_date() then 'Active' else 'Expired' end...	1000 row(s) returned	0.000 sec / 0.015 sec
15 06:32:38	select emp_no, dept_no, from_date from dept_manager where emp_no in (select emp_no from employees whe...	19 row(s) returned	0.031 sec / 0.000 sec

Object Info Session

Retrieve the employee ID, first name, and last name of employees who earn the highest salary. (1 marks)

MySQL Workbench

Local instance MySQL80 ×

File Edit View Query Database Server Tools Scripting Help

Navigator: employees HA_CLINICS_DB Janada_EOM* Janada_EOM_Part2*

SCHEMAS

Filter objects

- cooperativesdb
- customers
- employees
 - Tables
 - dept_emp
 - dept_manager
 - employees
 - salaries
 - titles
- Views

Administration Schemas

Information: No object selected

```
-- Retrieve the employee ID, first name, and last name of employees who earn the highest salary. (1 marks)
55  select e.emp_no, e.first_name, e.last_name
56  from employees e
57  join salaries s on e.emp_no = s.emp_no
58  where s.salary = (
59    select max(salary)
60    from salaries
61  )
```

Result Grid | Filter Rows: Export: Wrap Cell Content: Result 10 ×

emp_no	first_name	last_name
43624	Tokuyasu	Pesch

Action Output

Time	Action	Message	Duration / Fetch
14 06:28:17	select emp_no, salary, from_date, to_date, case when to_date > current_date() then 'Active' else 'Expired' end...	1000 row(s) returned	0.000 sec / 0.015 sec
15 06:32:38	select emp_no, dept_no, from_date from dept_manager where emp_no in (select emp_no from employees whe...	19 row(s) returned	0.031 sec / 0.000 sec
16 06:35:34	select e.emp_no, e.first_name, e.last_name from employees e join salaries s on e.emp_no = s.emp_no where s....	1 row(s) returned	1.891 sec / 0.000 sec

Object Info Session

NUMBER 7

Navigator: employees HA_CLINICS_DB* Janada_EOM* Janada_EOM_Part2*

SCHEMAS

Filter objects

- cooperativesdb
- customers
- employees
 - Tables
 - dept_emp
 - dept_manager
 - employees
 - functions
 - hope_alive_clinics
 - Views
- Stored Procedures
- Triggers

Administration Schemas

Information: Schema: employees

```
104  JOIN departments d
105    ON de.dept_no = d.dept_no
106    -- Department manager
107  JOIN dept_manager dm
108    ON d.dept_no = dm.dept_no
109    AND dm.to_date = '9999-01-01'
110  JOIN employees m
111    ON dm.emp_no = m.emp_no
112  ORDER BY e.emp_no;
```

Result Grid | Filter Rows: Export: Wrap Cell Content: Fetch rows: Result 1 ×

emp_no	first_name	last_name	gender	hire_date	current_title	current_salary	dept_name	manager_emp_no	manager_first_name	manager_last_name
10001	Georgi	Facello	M	1986-06-26	Senior Engineer	88958	Development	110567	Leon	DasSarma
10002	Bezalel	Simmel	F	1985-11-21	Staff	72527	Sales	111133	Hauke	Zhang
10003	Porto	Bamford	M	1986-08-28	Senior Engineer	43311	Production	110420	Oscar	Ghazale
10004	Christian	Koblick	M	1986-12-01	Senior Engineer	74057	Production	110420	Oscar	Ghazale
10005	Kyoko	Malinka	M	1989-09-12	Senior Staff	94692	Human Resources	110228	Karsten	Sigstam
10006	Anneke	Preusig	F	1989-06-02	Senior Engineer	59755	Development	110567	Leon	DasSarma
10007	Tzvetan	Zielinski	F	1989-02-10	Senior Staff	88070	Research	111534	Hilary	Kambil
10009	Sumant	Peac	F	1985-02-18	Senior Engineer	94409	Quality Management	110854	Dung	Pesch
10010	Duangkaew	Piveteau	F	1989-08-24	Engineer	80324	Quality Management	110854	Dung	Pesch
10012	Patricia	Bridgland	M	1992-12-18	Senior Engineer	54423	Development	110567	Leon	DasSarma
10013	Eberhardt	Terkki	M	1985-10-20	Senior Staff	68901	Human Resources	110228	Karsten	Sigstam
10014	Berni	Genin	M	1987-03-11	Engineer	60598	Development	110567	Leon	DasSarma
10016	Kazuhito	Cappelletti	M	1995-01-27	Staff	77935	Sales	111133	Hauke	Zhang

Action Output

Time	Action	Message	Duration / Fetch
1 20:35:50	SELECT e.emp_no, e.first_name, e.last_name, e.gender, e.hire_date, t.title AS current_ttl...	Error Code: 1146: Table 'hope_alive_clinics.employees' doesn't exist	0.000 sec
2 20:36:01	SELECT e.emp_no, e.first_name, e.last_name, e.gender, e.hire_date, t.title AS current_ttl...	1000 row(s) returned	10.141 sec / 0.000 sec