Jawaban Soal 1: Perancangan Struktur Database

1. Bentuk Tidak Ter-Normalisasi (UNF)

Client ID	Client Name	Business Type	Project Name	Project Budget	Invoice Number	Invoice Amount
1	ABC Corp	Retail	Project A	100,000.00	INV001	20,000.00
1	ABC Corp	Retail	Project A	100,000.00	INV002	80,000.00
2	XYZ Ltd	Technology	Project B	150,000.00	INV003	150,000.00

2. Bentuk Normal Pertama (1NF)

Hasil normalisasi 1NF:

Client ID	Client Name	Business Type	Project ID	Project Name	Project Budget	Invoice ID	Invoice Number	Invoice Amount
1	ABC Corp	Retail	1	Project A	100,000.00	1	INV001	20,000.00
1	ABC Corp	Retail	1	Project A	100,000.00	2	INV002	80,000.00
2	XYZ Ltd	Technology	2	Project B	150,000.00	3	INV003	150,000.00

3. Bentuk Normal Kedua (2NF)

Tabel: Clients

Client ID	Client Name	Business Type
1	ABC Corp	Retail
2	XYZ Ltd	Technology

Tabel: Projects

Project ID	Client ID	Project Name	Project Budget
1	1	Project A	100,000.00
2	2	Project B	150,000.00

Tabel: Invoices

Invoice ID	Project ID	Invoice Number	Invoice Amount
1	1	INV001	20,000.00
2	1	INV002	80,000.00
3	2	INV003	150,000.00

4. Bentuk Normal Ketiga (3NF)

Menghilangkan ketergantungan transitif, sehingga data hanya bergantung pada primary key masing-masing tabel.

Hasil normalisasi pada 3NF terdiri dari tabel berikut:

Tabel: Clients

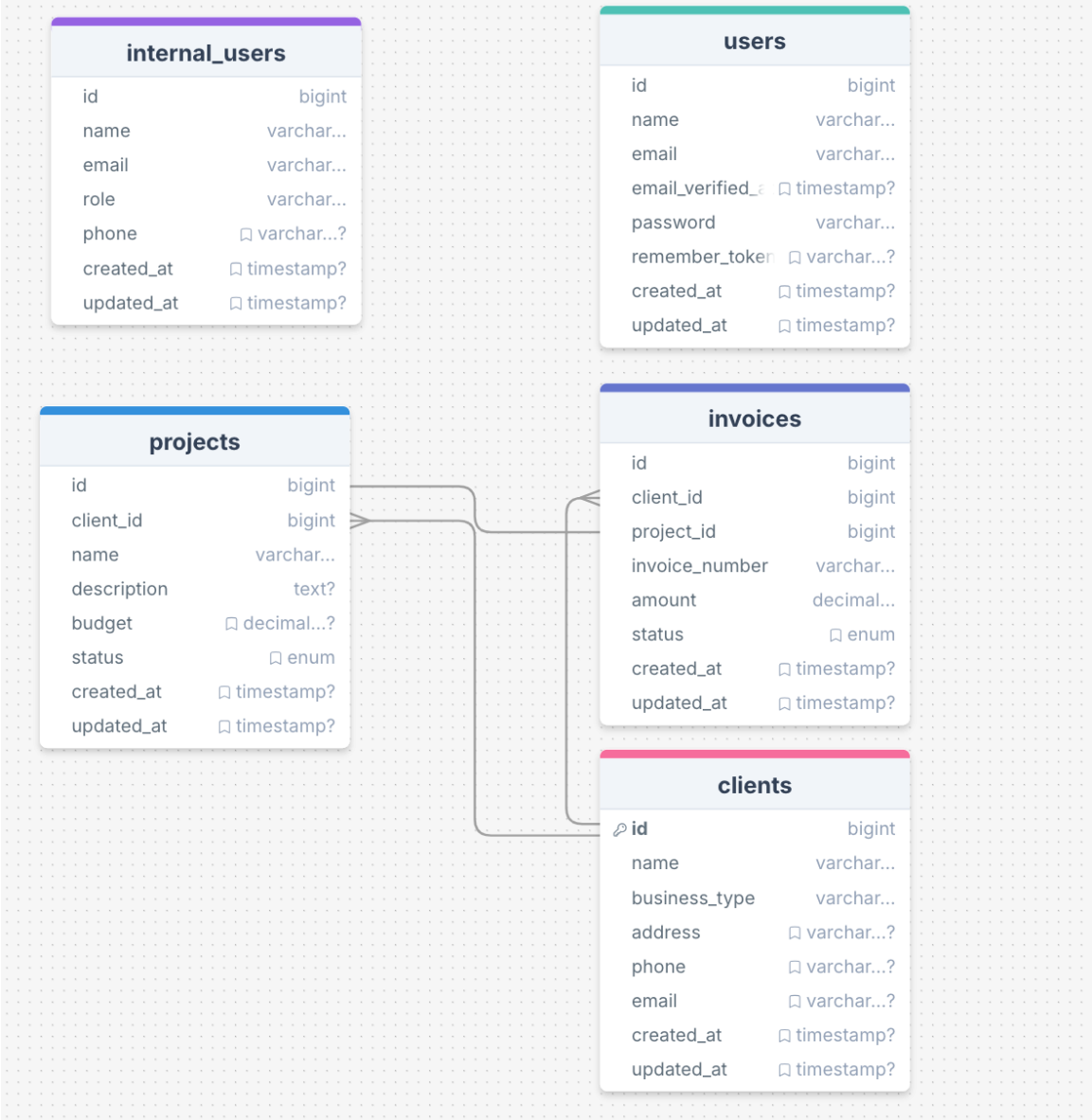
Client ID	Client Name	Business Type
1	ABC Corp	Retail
2	XYZ Ltd	Technology

Tabel: Projects

Project ID	Client ID	Project Name	Project Budget
1	1	Project A	100,000.00
2	2	Project B	150,000.00

Tabel: Invoices

Invoice ID	Project ID	Invoice Number	Invoice Amount
1	1	INV001	20,000.00
2	1	INV002	80,000.00
3	2	INV003	150,000.00


internal_users
id bigint
name varchar...
email varchar...
role varchar...
phone varchar...?
created_at timestamp?
updated_at timestamp?
users
id bigint
name varchar...
email varchar...
email_verified_a timestamp?
password varchar...
remember_token varchar...?
created_at timestamp?
updated_at timestamp?
projects
id bigint
client_id bigint
name varchar...
description text?
budget decimal...?
status enum
created_at timestamp?
updated_at timestamp?
invoices
id bigint
client_id bigint
project_id bigint
invoice_number varchar...
amount decimal...
status enum
created_at timestamp?
updated_at timestamp?
clients
id bigint
name varchar...
business_type varchar...
address varchar...?
phone varchar...?
email varchar...?
created_at timestamp?
updated_at timestamp?

Jawaban Soal 2: Query SQL

```sql
SELECT
    invoices.invoice_number,
    invoices.amount,
    invoices.status AS invoice_status,
    projects.name AS project_name,
    clients.name AS client_name,
    clients.email AS client_email
FROM invoices
JOIN projects ON invoices.project_id = projects.id
JOIN clients ON invoices.client_id = clients.id;
```

100% 39:4

Result Grid Filter Rows: Search Export:

invoice_numb...	amount	invoice_stat...	project_name	client_name	client_email
INV-001	5000.00	unpaid	Website Development	Alpha Corp	contact@alphacorp.com
INV-002	3000.00	paid	Market Analysis	Beta Solutions	info@betasolutions.com
INV-003	10000.00	unpaid	Product Design	Gamma Industries	support@gammaind.com
INV-004	2000.00	paid	Financial Audit	Delta Ventures	hello@deltaventures.com
INV-005	7500.00	unpaid	Retail Expansion	Epsilon Enterprises	sales@epsilon.com

Penjelasan: Query ini menggabungkan tabel invoices, projects, dan clients menggunakan kolom foreign key. Hasilnya adalah daftar invoice lengkap dengan nama proyek dan klien terkait.

```sql
SELECT
    clients.name AS client_name,
    clients.email,
    (SELECT SUM(projects.budget)
     FROM projects
     WHERE projects.client_id = clients.id) AS total_budget
FROM clients;
```

100% 14:7

Result Grid Filter Rows: Search Export:

client_name	email	total_budg...
Alpha Corp	contact@alphacorp.com	5000.00
Beta Solutions	info@betasolutions.com	3000.00
Gamma Industries	support@gammaind.com	10000.00
Delta Ventures	hello@deltaventures.com	2000.00
Epsilon Enterprises	sales@epsilon.com	7500.00

Penjelasan: Subquery menghitung total anggaran proyek setiap klien menggunakan SUM pada tabel projects. Hasilnya ditampilkan bersama data klien.

Jawaban Soal 3: Implementasi Database Object

1. **View**

```
1 •  SELECT * FROM `uas-database`.client_projects_summary;
```

100% 54:1

Result Grid | Filter Rows: Search | Export:

client_name	total_projects	total_budg...
Alpha Corp	1	5000.00
Beta Solutions	1	3000.00
Gamma Industries	1	10000.00
Delta Ventures	1	2000.00
Epsilon Enterprises	1	7500.00

Kasus Penggunaan: View ini memberikan ringkasan jumlah proyek dan total anggaran untuk setiap klien.

2. **Procedure**

```
DELIMITER $$
CREATE PROCEDURE update_invoice_status(IN invoiceID BIGINT, IN newStatus ENUM('unpaid', 'paid', 'overdue'))
BEGIN
    UPDATE invoices
    SET status = newStatus, updated_at = NOW()
    WHERE id = invoiceID;
END $$
DELIMITER ;
```

Kasus Penggunaan: Procedure ini digunakan untuk memperbarui status invoice dengan cepat.

3. **Trigger**

```
DELIMITER $$
CREATE TRIGGER update_project_status
AFTER UPDATE ON invoices
FOR EACH ROW
BEGIN
    IF NEW.status = 'paid' THEN
        UPDATE projects
        SET status = 'completed', updated_at = NOW()
        WHERE id = NEW.project_id;
    END IF;
END $$
DELIMITER ;
```

Kasus Penggunaan: Trigger ini otomatis memperbarui status proyek menjadi completed ketika semua invoice terkait telah paid.