CIS344 Project 1 — Freelance Graphic Design Studio

MD MOIN UDDIN SAGOR’S Freelance Graphic Design Studio
Muhammadmoinuddin13@gmail.com

This repo contains a MySQL database project for a freelance graphic design studio: clients, projects, services, tasks, time tracking, files, revisions, expenses, estimates, invoices, payments, and communications.

Files in this repository

1. CIS344_design studio.sql — Run this in MySQL Workbench to create the schema and load demo data.
2. CIS344_ER_design_studio.mwb` — MySQL Workbench model (UML/EER diagram).
3. CIS344_hand-drawn Chen diagram .pdf` — Chen-style ER diagram (hand drawn).
4. docs:ERD_UML.png` — Exported UML/EER diagram image (PNG).
5. Screenshots of query outputs:
   a. CIS344_SS of SQL output 1.png
   b. CIS344_SS of SQL output 2.png
   c. CIS344_SS of SQL output 3.png
   d. CIS344_SS of SQL output 4.png

How to run (MySQL Workbench)

1. Open Workbench → File → Open SQL Script
   Select CIS344_design studio.sql` and click the lightning bolt to execute.
2. Reverse-engineer the schema to view/edit the UML diagram: Database → Reverse Engineer… → choose your connection →

select the design_studio schema → finish. Save the model as needed (you already have CIS344_ER_design_studio.mwb).

3. Export diagram (if requested): From the EER Diagram window: File → Export → PNG/PDF (already provided as docs: ERD_UML.png).

Run examples:

```sql
SELECT * FROM v_client_balances ORDER BY balance_due DESC;
SELECT * FROM v_project_profitability ORDER BY profit_estimate DESC;
```

System Requirements & Process — Freelance Graphic Design Studio

MD MOIN UDDIN SAGOR’S Freelance Graphic Design Studio
Muhammadmoinuddin13@gmail.com

1) Project Overview & Scope

Design a relational database to run a small freelance graphic design studio: manage clients, projects, services, tasks, time tracking, files, revisions, expenses, estimates, invoices, payments, and communications. Support quoting, billing (fixed/hourly), tracking deliverables/revisions, and reporting on balances and profitability.

2) Stakeholders & Roles

- Studio Owner (Admin): creates clients/projects, sets rates, issues invoices, records payments, runs reports.
- Designer/Contractor: logs time, updates task status, uploads deliverables.
- Client (External): receives estimates/invoices, requests revisions, provides feedback.
- Bookkeeper (Optional): reconciles payments, views financial reports.

3) Functional Requirements (FR)

- FR-01 Clients — manage client info (name, email, phone, billing address).
- FR-02 Services & Rates — maintain service catalog with default hourly rates.
- FR-03 Projects — per client; billing type (hourly/fixed), dates, budget.
- FR-04 Project Services — attach services with quantity + rate snapshot.
- FR-05 Tasks & Assignment — tasks with estimate hours, priority/status, assigned user.
- FR-06 Time Tracking — log billable/non-billable time; duration auto-calculated.
- FR-07 Files/Deliverables — store links/metadata; optional task link; version.
- FR-08 Revision Rounds — track revision number per project/task with notes.
- FR-09 Expenses — record project expenses; mark billable.
- FR-10 Estimates/Quotes — estimates with line items; status (draft/sent/accepted/rejected).
- FR-11 Invoicing — invoices with line items from time/expenses/services; status (draft/sent/paid/void).
- FR-12 Payments — record payments; compute remaining balance.
- FR-13 Communications Log — log calls/emails/meetings tied to client/project/user.
- FR-14 Reporting — client balances and project profitability (views provided).
- FR-15 Audit/History — snapshot rates on items/time; avoid destructive cascades that lose history.

4) Non-Functional Requirements (NFR)

- NFR-01 Data Integrity: PKs/FKs, checks (end_time > start_time), unique constraints (revision per project).
- NFR-02 Security: least privilege; minimal PII.
- NFR-03 Availability: desktop/lab use; backup via SQL export.
- NFR-04 Performance: queries return <1s on lab data; indexes on FKs.
- NFR-05 Usability: clear naming; views for common reports; seed data for demo.
- NFR-06 Compliance: no card numbers; only necessary client data.

5) Business Rules

1. Client 1—N Project; Project belongs to exactly one Client.
2. N—M Project–Service via project_service(project_id, service_id) with rate snapshot.
3. Project 1—N Task; Task may be assigned to one user (nullable).
4. TimeEntry requires user + project; task optional; end_time > start_time.
5. Revision.rev_number unique per project.
6. Estimate/Invoice totals = sum(items) + tax; status lifecycle enforced.
7. Sum(payments) ≤ invoice.total; invoice is "paid" when covered.
8. Billable expenses can be invoiced; non-billable are internal cost.
9. Deleting a Project cascades to child records, not to Client.

6) Data Model (Entities → Key Attributes)

client; project; service; project_service; user_account; task; time_entry; file_asset; revision; expense; estimate + estimate_line_item; invoice + invoice_line_item; payment; communication_log. (Matches your SQL.)

7) Research, Interviews, and Surveys

Interview Guides

- **Studio Owner/Admin**
    1. What services do you sell most often? Fixed vs hourly?
    2. What info do you always need on clients/projects?
    3. How do you create estimates/invoices now? Any approvals?
    4. What counts as a "task" vs "deliverable"?
    5. How do you track revision rounds/out-of-scope work?
    6. Which reports do you need weekly/monthly?
    7. What records must be kept for tax/audits?
    8. Pain points with time tracking/contractors?
- **Designer/Contractor**
    1. Minimum fields needed for time entries?
    2. Do you switch tasks often in a day?
    3. What deliverables do you upload? How do you version?
    4. How do you capture client feedback?
    5. Useful task statuses?
    6. Which reminders/notifications help?
- **Client (if available)**
    1. What do you expect before work begins (brief, estimate)?
    2. Preferred invoice + payment method?
    3. How do you want to request revisions/approvals?
    4. Milestones/turnaround expectations?

Short Survey (Likert 1–5)

- "Invoice line items are easy to understand."
- "The number of revision rounds is sufficient."
- "Task statuses/due dates keep work on track."
- "Time entries accurately reflect work performed."
- "Reports give me the insights I need."

Synthesized Findings (sample)

- Need both fixed-fee and hourly with **rate snapshots**.
- Owners want **who owes what** and **which projects are profitable**.
- Designers want **simple** time logging with auto-duration.
- Clients want **clear deliverable names** and visible **revision counts**.

8) Process Steps (what I did)

1. Gather requirements (interviews/survey + best-practice research).
2. Define entities & rules.
3. Draw Chen ERD for cardinalities/optionality.
4. Convert to UML/EER in MySQL Workbench.
5. Build DDL with constraints, indexes, and views.
6. Load seed data for demos.
7. Validate with queries/views.
8. Iterate based on gaps/feedback.