University of British Columbia, Vancouver
Department of Computer Science

CPSC 304 Project Cover Page

Milestone 2 - Logical Design, RS, SQL DDL, Normalization, Query Design

Date: 22-October-2025

Group Number: 11

Name	Student Number	CS Alias (Userid)	Preferred E-mail Address
Lalith Krishna Vangara	55947394	v5m5m	lalithkrishna.v@gmail.com
Ankit Aggarwal	69484194	h5d8p	ankit.acoolguy@gmail.com
Imanjosh Chahal	30664536	n3f7v	imanjoshchahal@gmail.com

By typing our names and student numbers in the above table, we certify that the work in the attached assignment was performed solely by those whose names and student IDs are included above. (In the case of Project Milestone 0, the main purpose of this page is for you to let us know your e-mail address, and then let us assign you to a TA for your project supervisor.)

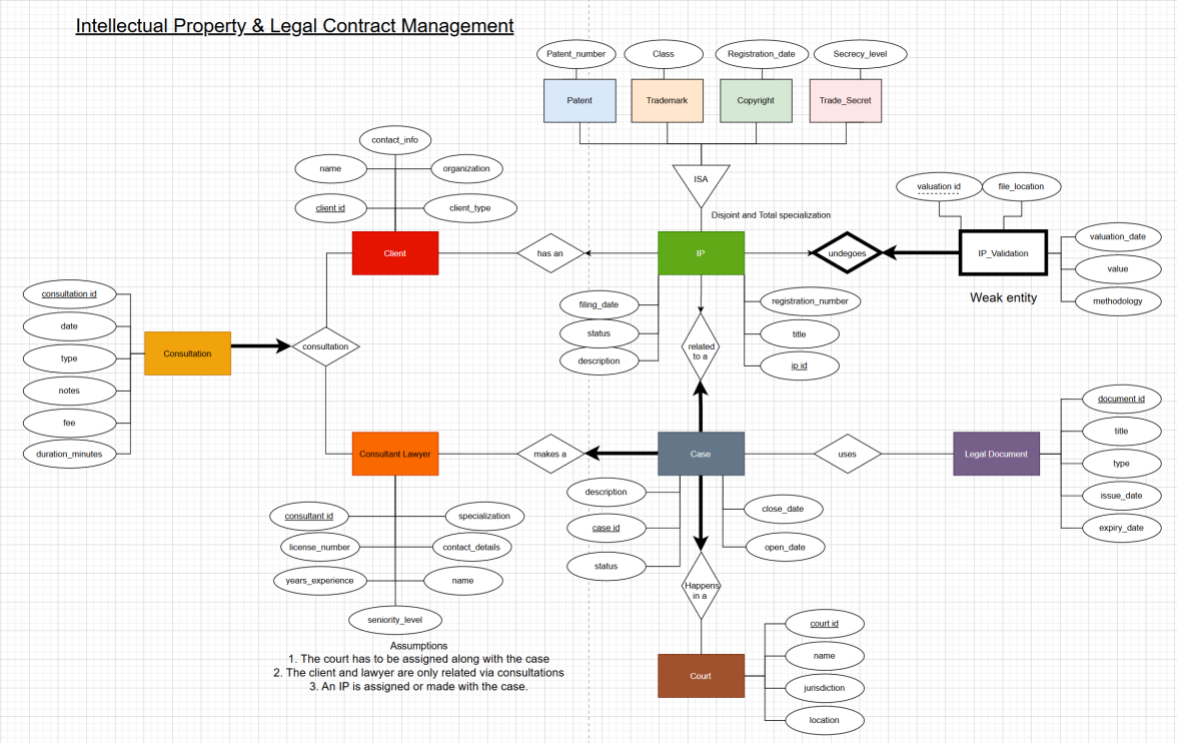
In addition, we indicate that we are fully aware of the rules and consequences of plagiarism, as set forth by the Department of Computer Science and the University of British Columbia

University of British Columbia, Vancouver
Department of Computer Science

Summary:

This project models an intellectual property (IP) legal management system. It represents the relationships between clients, consultant lawyers, legal cases, and various types of intellectual property (patents, trademarks, copyrights, and trade secrets).

ER Diagram:



Changes Made Since Milestone 1:

1. **Relationship Simplification:**
   The two binary relationships among *Client*, *Consultant Lawyer*, and *Consultation* were combined into a **single ternary relationship** for better data integrity and clarity.
2. **Attribute Additions for Functional Dependencies (FDs):**
   - duration_minutes was added to the **Consultation** entity to better capture consultation metrics and enable more FDs.
   - seniority_level was added to **Consultant_Lawyer** to reflect expertise and define FDs based on experience.

University of British Columbia, Vancouver
Department of Computer Science

Schema and Functional Dependencies (FD's):

1. Patent(patent_number: VARCHAR(100), ip_id: INTEGER)

- **Primary Key (PK)**: ip_id
- **Foreign Key (FK)**: ip_id
- **Functional Dependencies (FDs)**:
  - ip_id → patent_number
  - patent_number → ip_id
- **Candidate Key**: patent_number

2. Trademark(class: VARCHAR(255), ip_id: INTEGER)

- **PK**: ip_id
- **FK**: ip_id
- **FD**: ip_id → class

3. Copyright(registration_date: DATE, ip_id: INTEGER)

- **PK**: ip_id
- **FK**: ip_id
- **FD**: ip_id → registration_date

4. Trade_Secret(secrecy_level: VARCHAR(100), ip_id: INTEGER)

- **PK**: ip_id
- **FK**: ip_id
- **FD**: ip_id → secrecy_level

5. IP_merge_has_an(ip_id: INTEGER, filing_date: DATE, status: VARCHAR(50), description: CLOB, title: VARCHAR(255), registration_number: VARCHAR(100), client_id: INTEGER)

- **PK**: ip_id
- **Candidate Key**: registration_number
- **FDs**:
  - ip_id → filing_date, status, description, title, registration_number, client_id
  - registration_number → ip_id, filing_date, status, description, title, client_id

- **FK**: client_id

6. IP_Validation(ip_id: INTEGER, valuation_id: INTEGER, file_location: VARCHAR(255), valuation_date: DATE, value: DECIMAL(15, 2), methodology: CLOB)

- **PK**: (ip_id, valuation_id)
- **FD**: (ip_id, valuation_id) → file_location, valuation_date, value, methodology
- **FK**: ip_id

7. Client(client_id: INTEGER, name: VARCHAR(255), contact_info: VARCHAR(255), organization: VARCHAR(255), client_type: VARCHAR(100))

- **PK**: client_id
- **Candidate Key**: (name, contact_info)
- **FDs**:
    - client_id → name, contact_info, organization, client_type
    - (name, contact_info) → client_id, organization, client_type
    - organization → client_type

8. Case(case_id: INTEGER, consultant_id: INTEGER, ip_id: INTEGER, court_id: INTEGER, description: CLOB, status: VARCHAR(50), open_date: DATE, close_date: DATE)

- **PK**: case_id
- **FKs**: consultant_id, ip_id (unique), court_id (all are not null)
- **Candidate Key**: ip_id
- **FDs**:
    - case_id → consultant_id, ip_id, court_id, description, status, open_date, close_date
    - ip_id → case_id *(ip_id is unique)*

9. Court(court_id: INTEGER, name: VARCHAR(255), jurisdiction: VARCHAR(255), location: VARCHAR(255))

- **PK**: court_id
- **FDs**:
    - court_id → name, jurisdiction, location
    - jurisdiction → location

---

10. Legal_Document(document_id: INTEGER, title: VARCHAR(255), type: VARCHAR(100), issue_date: DATE, expiry_date: DATE)

- **PK**: document_id
- **FDs**:
    - document_id → title, type, issue_date, expiry_date
    - (type, issue_date) → expiry_date

11. Consultant_Lawyer(consultant_id: INTEGER, name: VARCHAR(255), license_number: VARCHAR(100), years_experience: INTEGER, specialization: VARCHAR(255), contact_details: VARCHAR(255), seniority_level: VARCHAR(100))

- **PK**: consultant_id
- **Candidate Keys**: license_number, (name, contact_details)
- **FDs**:
    - consultant_id → name, license_number, years_experience, specialization, contact_details, seniority_level
    - license_number → consultant_id, name, years_experience, specialization, contact_details, seniority_level
    - (name, contact_details) → consultant_id, license_number, years_experience, specialization, seniority_level
    - years_experience → seniority_level

12. Uses(case_id: INTEGER, document_id: INTEGER)

- **PK**: (case_id, document_id)
- **FD**: none
- **FKs**: case_id, document_id

13. Consultation(consultation_id: INTEGER, client_id: INTEGER, consultant_id: INTEGER, date: DATETIME, type: VARCHAR(100), notes: CLOB, fee: DECIMAL(10, 2), duration_minutes: INTEGER)

- **PK**: consultation_id
- **FKs**: client_id, consultant_id
- **FDs**:

- consultation_id → client_id, consultant_id, date, type, notes, fee, duration_minutes
- type → fee
- type → duration_minutes
- duration_minutes → fee

Normalization (BCNF; based on the above schema and FDs):

1. Because the LHS of both the FDs above is a superkey, the relation is already in BCNF. Therefore, no decomposition required.
2. Because the LHS of the above FD is a superkey, the relation is already in BCNF. Therefore, no decomposition required.
3. Because the LHS of the above FD is a superkey, the relation is already in BCNF. Therefore, no decomposition required.
4. Because the LHS of the above FD is a superkey, the relation is already in BCNF. Therefore, no decomposition required.
5. Because the LHS of both the FDs above is a superkey, the relation is already in BCNF. Therefore, no decomposition required.
6. Because the LHS of the above FD is a superkey, the relation is already in BCNF. Therefore, no decomposition required.
7. Not in BCNF, and therefore a BCNF decomposition is required…

Decomposing on (organization → client_type):

Organization_Type(organization: VARCHAR(255), client_type: VARCHAR(100)) &

Client(client_id: INTEGER, name: VARCHAR(255), contact_info: VARCHAR(255), organization: VARCHAR(255))

Both the relations above are now in BCNF.

8. Because the LHS of both the FDs above is a superkey, the relation is already in BCNF. Therefore, no decomposition required.
9. Not in BCNF, and therefore a BCNF decomposition is required…

Decomposing on (jurisdiction → location):

Jurisdiction(jurisdiction: VARCHAR(255), location: VARCHAR(255)) &

Court(court_id: INTEGER, name: VARCHAR(255), jurisdiction: VARCHAR(255))

Both the relations above are now in BCNF.

10. Not in BCNF, and therefore a BCNF decomposition is required…

Decomposing on ((type, issue_date) → expiry_date):

DocTypeIssue(type: VARCHAR(100), issue_date: DATE, expiry_date: DATE) &

Legal_Document(document_id: INTEGER, title: VARCHAR(255), type: VARCHAR(100), issue_date: DATE)

Both the relations above are now in BCNF.

11. Not in BCNF, and therefore a BCNF decomposition is required…

Decomposing on (years_experience → seniority_level):

Seniority(years_experience: INTEGER, seniority_level: VARCHAR(100)) &

Consultant_Lawyer(consultant_id: INTEGER, name: VARCHAR(255), license_number: VARCHAR(100), years_experience: INTEGER, specialization: VARCHAR(255), contact_details: VARCHAR(255))

Both the relations above are now in BCNF.

12. There are no FDs to violate BCNF, and hence, the relation is already in BCNF. Therefore, no decomposition required.
13. Not in BCNF, and therefore a BCNF decomposition is required…

Decomposing on (duration_minutes → fee):

DurationFee(duration_minutes, fee) & **Consultation_minusFee**(consultation_id, client_id, consultant_id, date, type, notes, duration_minutes)

Now decomposing Consultation_minusFee on (type → duration_minutes):

ConsultationType(type, duration_minutes) & **ConsultationBase**(consultation_id, client_id, consultant_id, date, type, notes)

Final BCNF relations:

ConsultationBase(consultation_id: INTEGER, client_id: INTEGER, consultant_id: INTEGER, date: DATE, type: VARCHAR(100), notes: CLOB) &

ConsultationType(type: VARCHAR(100), duration_minutes: INTEGER) &

DurationFee(duration_minutes: INTEGER, fee: DECIMAL(10, 2))

ALL tables post-normalization:

1. Patent(patent_number: VARCHAR(100), ip_id: INTEGER)

- **Primary Key (PK)**: ip_id
- **Foreign Key (FK):** ip_id
- **Candidate Key**: patent_number

2. Trademark(class: VARCHAR(255), ip_id: INTEGER)

- **PK**: ip_id
- **FK:** ip_id

3. Copyright(registration_date: DATE, ip_id: INTEGER)

- **PK**: ip_id
- **FK:** ip_id

4. Trade_Secret(secrecy_level: VARCHAR(100), ip_id: INTEGER)

- **PK**: ip_id
- **FK:** ip_id

5. IP_merge_has_an(ip_id: INTEGER, filing_date: DATE, status: VARCHAR(50), description: CLOB, title: VARCHAR(255), registration_number: VARCHAR(100), client_id: INTEGER)

- **PK**: ip_id
- **Candidate Key**: registration_number
- **Foreign Key (FK)**: client_id

---

6. IP_Validation(ip_id: INTEGER, valuation_id: INTEGER, file_location: VARCHAR(255), valuation_date: DATE, value: DECIMAL(15, 2), methodology: CLOB)

- **PK**: (ip_id, valuation_id)
- **FK**: ip_id

7. Organization_Type(organization: VARCHAR(255), client_type: VARCHAR(100))

- **PK:** organization

8. Client(client_id: INTEGER, name: VARCHAR(255), contact_info: VARCHAR(255), organization: VARCHAR(255))

- **PK:** client_id
- **Candidate Key:** (name,contact_info)
- **FK:** organization

9. Case(case_id: INTEGER, consultant_id: INTEGER, ip_id: INTEGER, court_id: INTEGER, description: CLOB, status: VARCHAR(50), open_date: DATE, close_date: DATE)

- **PK**: case_id
- **FKs**: consultant_id, ip_id (unique), court_id (all are not null)
- **Candidate Key**: ip_id

10. Jurisdiction(jurisdiction: VARCHAR(255), location: VARCHAR(255))

- **PK:** jurisdiction

11. Court(court_id: INTEGER, name: VARCHAR(255), jurisdiction: VARCHAR(255))

- **PK:** court_id
- **FK:** jurisdiction

12. DocTypeIssue(type: VARCHAR(100), issue_date: DATE, expiry_date: DATE)

- **PK:** (type, issue_date)

13. Legal_Document(document_id: INTEGER, title: VARCHAR(255), type: VARCHAR(100), issue_date: DATE)

- **PK:** document_id
- **FK:** (type, issue_date)

14. Seniority(years_experience: INTEGER, seniority_level: VARCHAR(100))

- **PK:** years_experience

15. Consultant_Lawyer(consultant_id: INTEGER, name: VARCHAR(255), license_number: VARCHAR(100), years_experience: INTEGER, specialization: VARCHAR(255), contact_details: VARCHAR(255))

- **PK:** consultant_id
- **FK:** years_experience
- **Candidate Keys:** license_number, (name, contact_details)

16. Uses(case_id: INTEGER, document_id: INTEGER)

- **PK**: (case_id, document_id)
- **FKs**: case_id, document_id

17. ConsultationBase(consultation_id: INTEGER, client_id: INTEGER, consultant_id: INTEGER, date: DATE, type: VARCHAR(100), notes: CLOB)

- **PK:** consultation_id
- **FK:** type, client_id, consultant_id

18. ConsultationType(type: VARCHAR(100), duration_minutes: INTEGER)

- **PK:** type
- **FK:** duration_minutes

19. DurationFee(duration_minutes: INTEGER, fee: DECIMAL(10, 2))

- **PK:** duration_minutes

University of British Columbia, Vancouver
Department of Computer Science

Find the SQL DDL and INSERT statements in the .sql file

Ai Declaration:

We did not use ai in this milestone of the project.