BCIT – COMP 2714

Relational Database Systems

Milestone 5 – Operational Database & Reporting Layer

LabTracker Project – Group 16

Fall 2025

Submitted by:

- ***Emanuel Molla***
- ***Jimmy Cho***
- ***Allen Rosales***
- ***Anthony Herradura***

Instructor:

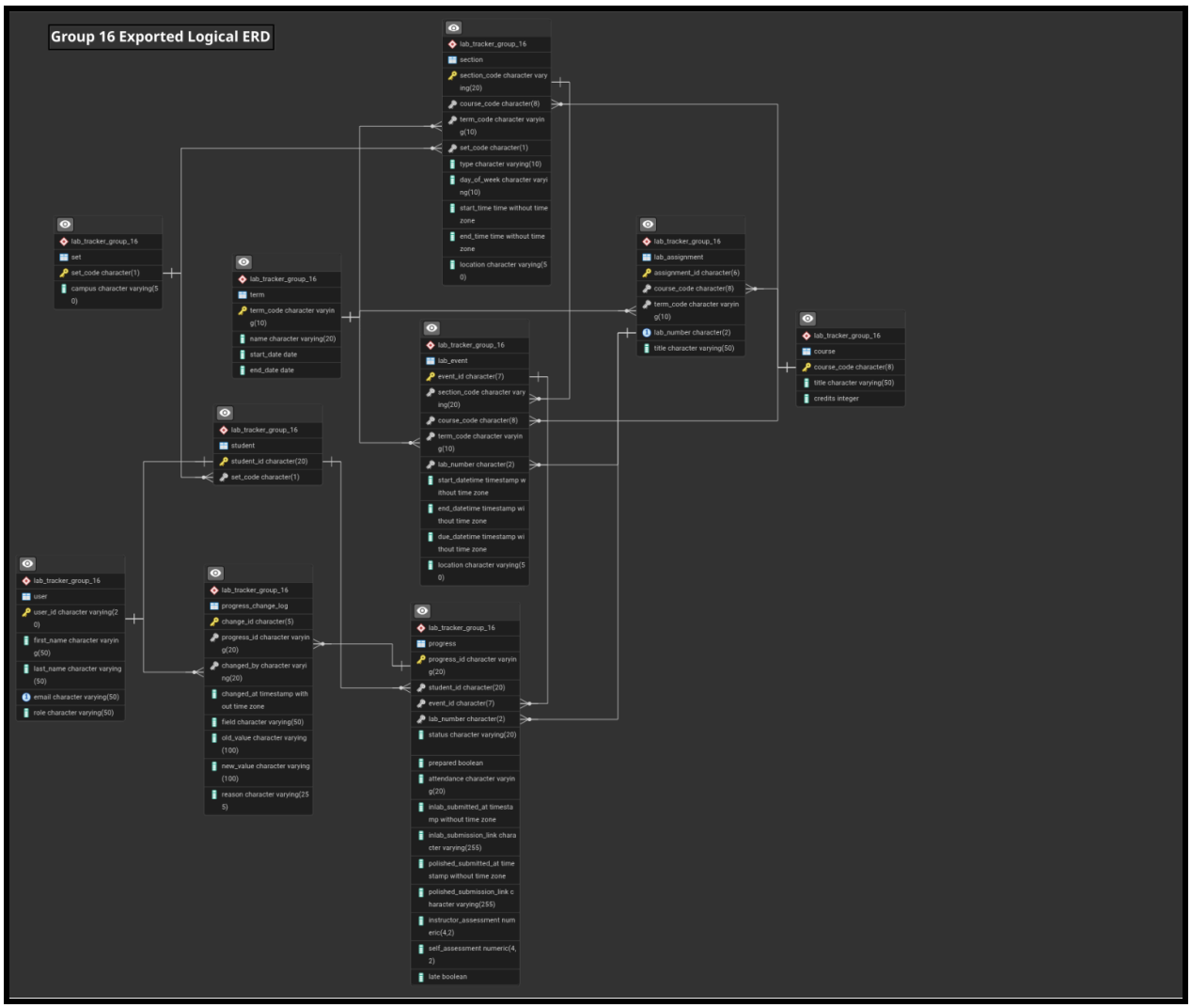
Maryam Khezrzadeh

Date Submitted:

November 30, 2025

FINAL DOCUMENTATION

UPDATED ERD



OBJECT LIST

The following objects represent all tables, views, roles, stored functions, and triggers used in the LabTracker schema for Milestone 5.

Table:

1) User
2) Course
3) Term
4) Set
5) Student
6) Section
7) Lab_assignment
8) Lab_event
9) Progress
10) Progress_change_log

Views:

1) v_ta_progress_summary
2) v_section_overview

Roles:

1) admin_role
2) instructor_role
3) student_role
4) ta_role

Functions:

1) fn_log_progress_change
2) fn_create_lab_event_for_section

Function Triggers:

1) trg_log_progress_change

EXECUTION ORDER

1) **01_queries.sql**
2) **02_views.sql**
3) **03_access_control.sql**
4) **04_procedural.sql**
5) **05_transaction_demo.sql**

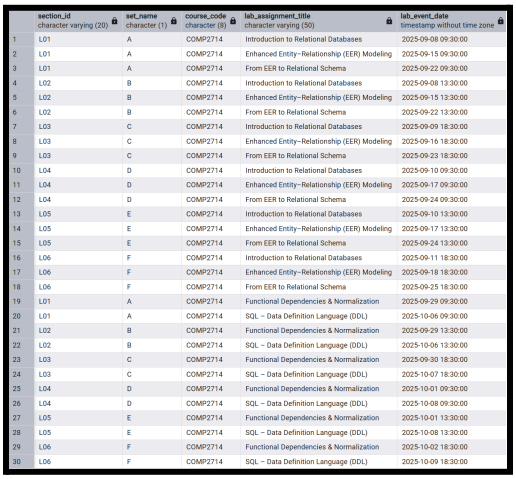
SCREENSHOTS

Part I : Queries screenshots

1. **Upcoming Lab Events** – List all lab_event rows for term `202530` (Fall 2025), showing: section_id, set name, course code, lab_assignment title, and lab_event date.

Query | Query History

```sql
SELECT
    le.section_code section_id,
    sec.set_code set_name,
    le.course_code course_code,
    la.title lab_assignment_title,
    le.start_datetime lab_event_date
FROM lab_event le
JOIN section sec
ON le.section_code = sec.section_code
JOIN lab_assignment la
ON la.lab_number  = le.lab_number
AND la.course_code = le.course_code
AND la.term_code   = le.term_code
WHERE le.term_code = '202530'
```

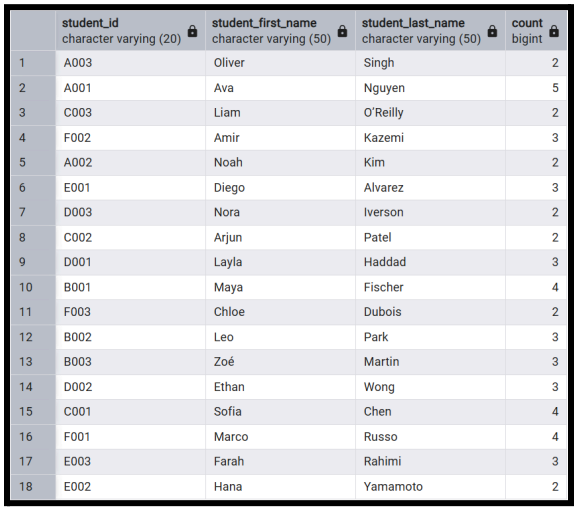


	section_id character varying (20)	set_name character (1)	course_code character (8)	lab_assignment_title character varying (50)	lab_event_date timestamp without time zone
1	L01	A	COMP2714	Introduction to Relational Databases	2025-09-08 09:30:00
2	L01	A	COMP2714	Enhanced Entity–Relationship (EER) Modeling	2025-09-15 09:30:00
3	L01	A	COMP2714	From EER to Relational Schema	2025-09-22 09:30:00
4	L02	B	COMP2714	Introduction to Relational Databases	2025-09-08 13:30:00
5	L02	B	COMP2714	Enhanced Entity–Relationship (EER) Modeling	2025-09-15 13:30:00
6	L02	B	COMP2714	From EER to Relational Schema	2025-09-22 13:30:00
7	L03	C	COMP2714	Introduction to Relational Databases	2025-09-09 18:30:00
8	L03	C	COMP2714	Enhanced Entity–Relationship (EER) Modeling	2025-09-16 18:30:00
9	L03	C	COMP2714	From EER to Relational Schema	2025-09-23 18:30:00
10	L04	D	COMP2714	Introduction to Relational Databases	2025-09-10 09:30:00
11	L04	D	COMP2714	Enhanced Entity–Relationship (EER) Modeling	2025-09-17 09:30:00
12	L04	D	COMP2714	From EER to Relational Schema	2025-09-24 09:30:00
13	L05	E	COMP2714	Introduction to Relational Databases	2025-09-10 13:30:00
14	L05	E	COMP2714	Enhanced Entity–Relationship (EER) Modeling	2025-09-17 13:30:00
15	L05	E	COMP2714	From EER to Relational Schema	2025-09-24 13:30:00
16	L06	F	COMP2714	Introduction to Relational Databases	2025-09-11 18:30:00
17	L06	F	COMP2714	Enhanced Entity–Relationship (EER) Modeling	2025-09-18 18:30:00
18	L06	F	COMP2714	From EER to Relational Schema	2025-09-25 18:30:00
19	L01	A	COMP2714	Functional Dependencies & Normalization	2025-09-29 09:30:00
20	L01	A	COMP2714	SQL – Data Definition Language (DDL)	2025-10-06 09:30:00
21	L02	B	COMP2714	Functional Dependencies & Normalization	2025-09-29 13:30:00
22	L02	B	COMP2714	SQL – Data Definition Language (DDL)	2025-10-06 13:30:00
23	L03	C	COMP2714	Functional Dependencies & Normalization	2025-09-30 18:30:00
24	L03	C	COMP2714	SQL – Data Definition Language (DDL)	2025-10-07 18:30:00
25	L04	D	COMP2714	Functional Dependencies & Normalization	2025-10-01 09:30:00
26	L04	D	COMP2714	SQL – Data Definition Language (DDL)	2025-10-08 09:30:00
27	L05	E	COMP2714	Functional Dependencies & Normalization	2025-10-01 13:30:00
28	L05	E	COMP2714	SQL – Data Definition Language (DDL)	2025-10-08 13:30:00
29	L06	F	COMP2714	Functional Dependencies & Normalization	2025-10-02 18:30:00
30	L06	F	COMP2714	SQL – Data Definition Language (DDL)	2025-10-09 18:30:00

2. **Student Participation Summary** – For each student, show how many lab events they attended (`attendance = TRUE`) based on student_progress.

Query | Query History

```sql
SELECT
    s.student_id,
    u.first_name student_first_name,
    u.last_name student_last_name,
    COUNT(*)
FROM student s
JOIN "user" u
ON u.user_id = s.student_id
LEFT JOIN progress p
ON p.student_id = s.student_id
WHERE p.attendance = 'Present'
GROUP BY s.student_id, u.first_name, u.last_name
```



	student_id character varying (20)	student_first_name character varying (50)	student_last_name character varying (50)	count bigint
1	A003	Oliver	Singh	2
2	A001	Ava	Nguyen	5
3	C003	Liam	O'Reilly	2
4	F002	Amir	Kazemi	3
5	A002	Noah	Kim	2
6	E001	Diego	Alvarez	3
7	D003	Nora	Iverson	2
8	C002	Arjun	Patel	2
9	D001	Layla	Haddad	3
10	B001	Maya	Fischer	4
11	F003	Chloe	Dubois	2
12	B002	Leo	Park	3
13	B003	Zoé	Martin	3
14	D002	Ethan	Wong	3
15	C001	Sofia	Chen	4
16	F001	Marco	Russo	4
17	E003	Farah	Rahimi	3
18	E002	Hana	Yamamoto	2

3. **Late Submissions** – List students who submitted late at least once (`late_flag = TRUE`), showing their set, section, and count of late submissions.

Query | Query History

```sql
SELECT
    s.student_id,
    u.first_name student_first_name,
    u.last_name student_last_name,
    s.set_code,
    le.section_code,
    COUNT(*) late_submission_count
FROM progress p
JOIN student s
ON p.student_id = s.student_id
JOIN "user" u
ON u.user_id = s.student_id
JOIN lab_event le
ON p.event_id = le.event_id
WHERE p.late = TRUE
GROUP BY
    s.student_id,
    u.first_name, u.last_name,
    s.set_code,
    le.section_code
HAVING COUNT(*) > 0
```

	student_id character varying (20)	student_first_name character varying (50)	student_last_name character varying (50)	set_code character (1)	section_code character varying (20)	late_submission_count bigint
1	A002	Noah	Kim	A	L01	1
2	B003	Zoé	Martin	B	L02	1
3	C003	Liam	O'Reilly	C	L03	1
4	E003	Farah	Rahimi	E	L05	1

4. **Instructor Assessment Report** – For each section, show the **average instructor_assessment** score across all student_progress records tied to that section's lab events.

Query | Query History

```sql
SELECT
    c.course_code course,
    s.section_code "section",
    ROUND(AVG(p.instructor_assessment), 2) AS avg_instructor_assessment
FROM course c
JOIN section s
ON s.course_code = c.course_code
LEFT JOIN lab_event le
ON le.section_code = s.section_code
LEFT JOIN progress p
ON p.event_id = le.event_id
GROUP BY s.section_code, c.course_code
ORDER BY s.section_code
```

	course character (8)	section character varying (20)	avg_instructor_assessment numeric
1	COMP2714	L01	8.61
2	COMP2714	L02	7.90
3	COMP2714	L03	7.85
4	COMP2714	L04	7.01
5	COMP2714	L05	8.91
6	COMP2714	L06	8.51

5. **Unassessed Progress** – List student_progress rows where **instructor_assessment IS NULL** or **self_assessment IS NULL**, including student name, section, and lab_event.

Query | Query History

```sql
SELECT
    p.progress_id,
    p.student_id,
    u.first_name student_first_name,
    u.last_name student_last_name,
    le.section_code "section",
    le.event_id lab_assignment_id,
    la.title lab_assignment_title,
    p.instructor_assessment,
    p.self_assessment
FROM progress p
JOIN student s
ON s.student_id = p.student_id
JOIN "user" u
ON u.user_id = s.student_id
JOIN lab_event le
ON le.event_id = p.event_id
JOIN lab_assignment la
ON la.lab_number  = p.lab_number
AND la.course_code = le.course_code
AND la.term_code   = le.term_code
WHERE p.instructor_assessment IS NULL
OR p.self_assessment IS NULL
```

	progress_id character varying (20)	student_id character (20)	student_first_name character varying (50)	student_last_name character varying (50)	section character varying (20)	lab_assignment_id character (7)	lab_assignment_title character varying (50)	instructor_assessment numeric (4,2)	self_assessment numeric (4,2)
1	A002-L01-L02	A002	Noah	Kim	L01	L01-L02	Enhanced Entity–Relationship (EER) Modeling	[null]	[null]
2	A003-L01-L03	A003	Oliver	Singh	L01	L01-L03	From EER to Relational Schema	[null]	[null]
3	B002-L02-L03	B002	Leo	Park	L02	L02-L03	From EER to Relational Schema	[null]	7.50
4	B002-L02-L02	B002	Leo	Park	L02	L02-L02	Enhanced Entity–Relationship (EER) Modeling	[null]	[null]
5	C001-L03-L04	C001	Sofia	Chen	L03	L03-L04	Functional Dependencies & Normalization	[null]	[null]
6	C002-L03-L03	C002	Arjun	Patel	L03	L03-L03	From EER to Relational Schema	4.00	[null]
7	C002-L03-L02	C002	Arjun	Patel	L03	L03-L02	Enhanced Entity–Relationship (EER) Modeling	[null]	[null]
8	D002-L04-L03	D002	Ethan	Wong	L04	L04-L03	From EER to Relational Schema	[null]	[null]
9	D002-L04-L02	D002	Ethan	Wong	L04	L04-L02	Enhanced Entity–Relationship (EER) Modeling	[null]	[null]
10	E002-L05-L03	E002	Hana	Yamamoto	L05	L05-L03	From EER to Relational Schema	[null]	[null]
11	E002-L05-L02	E002	Hana	Yamamoto	L05	L05-L02	Enhanced Entity–Relationship (EER) Modeling	[null]	[null]
12	F002-L06-L03	F002	Amir	Kazemi	L06	L06-L03	From EER to Relational Schema	7.80	[null]
13	F002-L06-L02	F002	Amir	Kazemi	L06	L06-L02	Enhanced Entity–Relationship (EER) Modeling	[null]	[null]
14	F003-L06-L03	F003	Chloe	Dubois	L06	L06-L03	From EER to Relational Schema	[null]	[null]

6. **Top Performers** – Find top students whose average instructor_assessment is ≥ 4.5, including their set and course/section info

Query | Query History

```sql
SELECT
    s.student_id,
    u.first_name student_first_name,
    u.last_name student_last_name,
    s.set_code,
    ROUND(AVG(p.instructor_assessment) / 2, 2) AS avg_instructor_assessment,
    e.course_code,
    e.section_code
FROM student s
JOIN "user" u
ON u.user_id = s.student_id
JOIN progress p
ON p.student_id = s.student_id
JOIN lab_event e
ON e.event_id = p.event_id
WHERE p.instructor_assessment IS NOT NULL
GROUP BY
    s.student_id,
    u.first_name,
    u.last_name,
    s.set_code,
    e.course_code,
    e.section_code
HAVING AVG(p.instructor_assessment) / 2 >= 4.5
ORDER BY avg_instructor_assessment DESC
```

	student_id character varying (20)	student_first_name character varying (50)	student_last_name character varying (50)	set_code character (1)	avg_instructor_assessment numeric	course_code character (8)	section_code character varying (20)
1	E003	Farah	Rahimi	E	4.83	COMP2714	L05
2	A003	Oliver	Singh	A	4.63	COMP2714	L01
3	C001	Sofia	Chen	C	4.60	COMP2714	L03

Part II : Views screenshots

1. **TA View Screenshot:** Provides TAs with a restricted view of student progress by hiding sensitive data (self assessments, timestamps, late flags).

```
SELECT * FROM v_ta_progress_summary LIMIT 10;
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS POSTGRESQL QUERY RESULTS

Results Messages

	section_id	lab_event_id	student_id	student_first_name	student_last_name	attendance	in_lab_submission_link	instructor_assessment
1	L01	L01-L02	A001	Ava	Nguyen	Present	https://submit.bcit.ca/comp2714/inlab/A001-L0…	7.00
2	L01	L01-L01	A001	Ava	Nguyen	Present	https://submit.bcit.ca/comp2714/inlab/A001-L0…	8.50
3	L01	L01-L02	A002	Noah	Kim	Present	https://submit.bcit.ca/comp2714/inlab/A002-L0…	NULL
4	L01	L01-L01	A002	Noah	Kim	Present	https://submit.bcit.ca/comp2714/inlab/A002-L0…	8.50
5	L01	L01-L02	A003	Oliver	Singh	Present	https://submit.bcit.ca/comp2714/inlab/A003-L0…	7.00
6	L01	L01-L01	A003	Oliver	Singh	Present	https://submit.bcit.ca/comp2714/inlab/A003-L0…	8.50
7	L02	L02-L02	B001	Maya	Fischer	Present	https://submit.bcit.ca/comp2714/inlab/B001-L0…	7.00
8	L02	L02-L01	B001	Maya	Fischer	Present	https://submit.bcit.ca/comp2714/inlab/B001-L0…	8.50
9	L02	L02-L02	B002	Leo	Park	Present	https://submit.bcit.ca/comp2714/inlab/B002-L0…	NULL
10	L02	L02-L01	B002	Leo	Park	Present	https://submit.bcit.ca/comp2714/inlab/B002-L0…	8.50

2. **Reporting View Screenshot:** Summarizes each section's total lab events and average instructor assessment for reporting purposes.

```
SELECT * FROM v_section_overview;
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS POSTGRESQL QUERY RESULTS

Results Messages

	term_code	set_name	course_code	section_id	total_events	avg_instructor_assessment
1	202530	A	COMP2714	L01	3	7.90
2	202530	B	COMP2714	L02	3	7.90
3	202530	C	COMP2714	L03	3	7.90
4	202530	D	COMP2714	L04	3	7.90
5	202530	E	COMP2714	L05	3	7.90
6	202530	F	COMP2714	L06	3	7.90

KNOWN ISSUES

1. **Limited Query Output Due to Seed Data**
   Some queries (e.g., late submissions, top performers, unassessed progress) show minimal or empty results because the seed data from previous milestones has limited coverage for the scenarios required in this milestone.

2. **User Creation Not Supported on Shared Server**
   The shared PostgreSQL server may block `CREATE USER` commands, so the `ta_demo` user may not actually be created. Role creation and GRANT statements still work as expected.

3. **ON CONFLICT Used to Avoid Collisions on Shared Environment**
   Since multiple groups share the same schema space, `ON CONFLICT DO NOTHING` is used in inserts for lab events and progress rows to prevent uniqueness conflicts. In rare cases, this may skip inserts if the ID already exists.

4. **Trigger Test Rows May Already Exist**
   The trigger testing section inserts a test progress row. If the same test ID exists from a previous run, the test must be cleaned up manually before re-running.

5. **Event ID Format Assumes Consistent Section and Lab Number Naming**
   The stored function builds event IDs using section code + lab number. This works for COMP2714's naming scheme, but would not generalize if section or lab numbering conventions change.