Errors File:

```sql
-- Course table
create table Course
(
    course_id int primary key identity (1,1),
    title nvarchar(30),
    category nvarchar(30),
    duration_hours int check (duration_hours > 0),
    levels varchar(20) check (levels in ('Beginner', 'Intermediate', 'Advanced'))


-- Trainee table
create table Trainee
```

100 %

Messages

Msg 102, Level 15, State 1, Line 12
Incorrect syntax near '}'.

Completion time: 2025-06-27T14:58:19.0990552+04:00

Missing addition of); after the last line

```sql
-- Insert data into Course
insert into Course (course_id, title, category, duration_hours, levels) value
(1, 'Database Fundamentals', 'Databases', 20, 'Beginner'),
(2, 'Web Development Basics', 'Web', 30, 'Beginner'),
(3, 'Data Science Introduction', 'Data Science', 25, 'Intermediate'),
(4, 'Advanced SQL Queries', 'Databases', 15, 'Advanced');


-- Insert data into Trainee
```

00 %

Messages

Msg 102, Level 15, State 1, Line 75
Incorrect syntax near 'value'.

Completion time: 2025-06-27T15:24:53.4199691+04:00

Writing values instead of value

```
  |  /,
  |  -- Trainer table
  |□create table Trainer
  |  (
  |     trainer_id int primary key identity (1,1),
  |     name nvarchar not null (30),
  |     specialty nvarchar(30),
  |     phone varchar(20),
  |     email varchar (30) unique
  |
  |  );
  |
  |  -- Schedule table
  |□create table Schedule
```

100 % ▾ ◀

▦ Messages

Msg 102, Level 15, State 1, Line 29
Incorrect syntax near '('.

Completion time: 2025-06-27T15:06:28.3327269+04:00

I ran into an issue because I needed the name column in the trainer table to be NOT NULL. To fix it, I dropped the trainer table first and then recreated it with the updated NOT NULL requirement for the name column. This way, no trainer can be added without a name.

```sql
    );

    drop table Trainer;

    create table Trainer
    (
        trainer_id int primary key identity (1,1),
        name nvarchar(30) not null,
        specialty nvarchar(30),
        phone nvarchar(20),
        email varchar(30) unique
    );

    -- Schedule table
```

100 %

Msg 2714, Level 16, State 6, Line 38
There is already an object named 'Trainer' in the database.

Completion time: 2025-06-27T15:16:20.7970284+04:00

I was trying to execute the newly created table after dropping it, but I got an error. Then I realized that each time I want to recreate the table, I need to run the DROP TABLE statement first, and then execute the NEW CREATE TABLE statement again.

```
    -- Insert data into Course
   insert into Course (course_id, title, category, duration_hours, levels) values
    (1, 'Database Fundamentals', 'Databases', 20, 'Beginner'),
    (2, 'Web Development Basics', 'Web', 30, 'Beginner'),
    (3, 'Data Science Introduction', 'Data Science', 25, 'Intermediate'),
    (4, 'Advanced SQL Queries', 'Databases', 15, 'Advanced');


    -- Insert data into Trainee
```

00 %   ▾ ◀

Messages

Msg 544, Level 16, State 1, Line 75
Cannot insert explicit value for identity column in table 'Course' when IDENTITY_INSERT is set to OFF.

Completion time: 2025-06-27T15:26:34.3631499+04:00

I discovered that inserting explicit values into an IDENTITY column causes an error because SQL Server does not allow manual insertion unless IDENTITY_INSERT is enabled. To fix this, I omitted the course_id from my insert statements so the database could automatically assign the IDs.
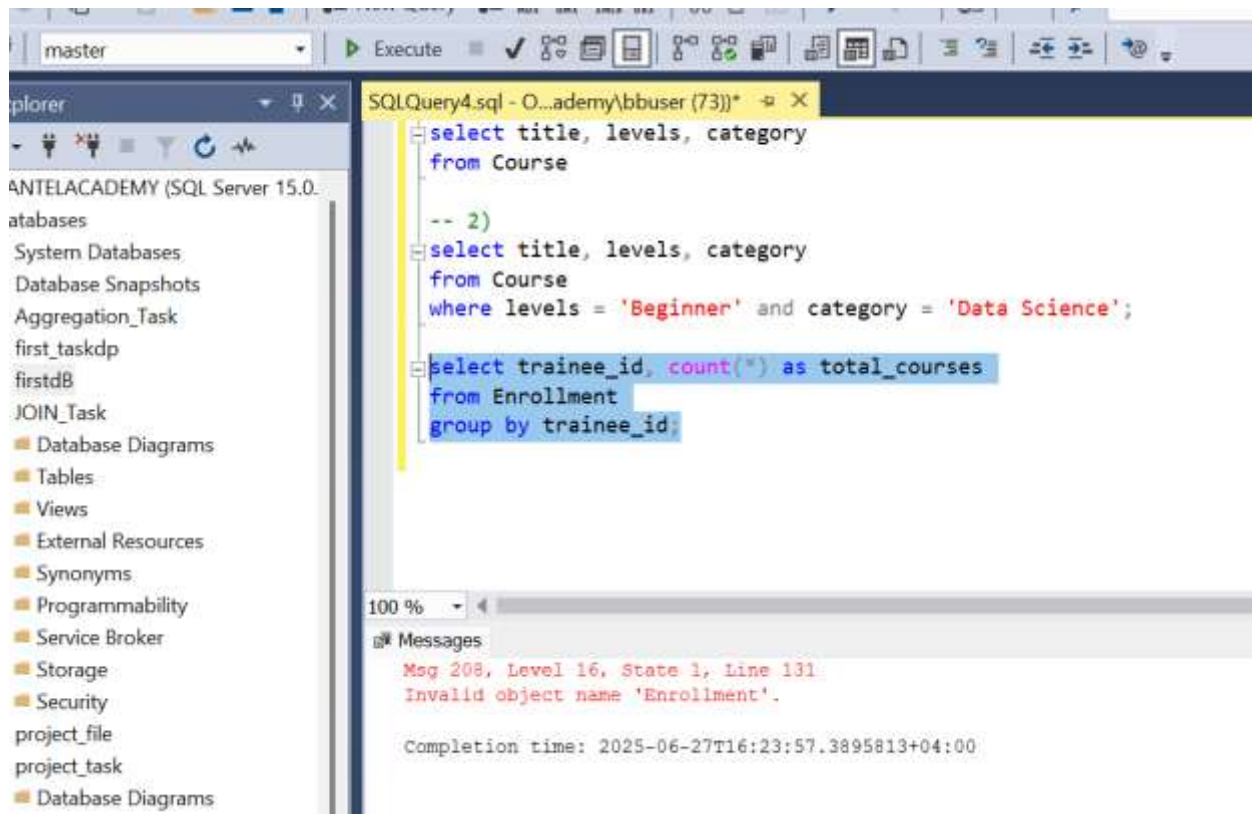
I was connected to the **master** database while trying to execute the code, so the table Enrollment was not recognized.

Solution: I switched from the **master** database to the correct one (**database_project**) using the USE command.

Msg 2714, Level 16, State 6, Line 6 There is already an object named 'Course' in the database.

Problem: The script didn't run correctly because some tables (like Course) already existed or were locked due to previous constraints or execution errors.

Solution: I opened a **new query window** and executed the full code again from the beginning, which worked successfully.

```
⊟, Course.title as course_title, Schedule.start_date Schedule.time_slot

    ainee_id = Trainee.trainee_id
    rse_id = Course.course_id
    e_id = Schedule.course_id
    art_date;
```

100 %   ▾ ◀

📧 Messages

    Msg 102, Level 15, State 1, Line 144
    Incorrect syntax near '.'.

    Completion time: 2025-06-28T12:16:15.2890379+04:00

I missed using (,) at the select line

```
-- 6)

select Trainee.name, Course.title, Trainer.name, Schedule.time_slot
from Enrollment
inner join Trainee on Enrollment.trainee_id = trainee.trainee_id
inner join Course on Eenrollment.course_id = course.course_id
inner join Schedule on Course.course_id = schedule.course_id
inner join Trainer on Schedule.trainer_id = trainer.trainer_id
```

0 %   ▾ ◀

Messages

    Msg 4104, Level 16, State 1, Line 162
    The multi-part identifier "Eenrollment.course_id" could not be bound.

    Completion time: 2025-06-28T12:36:23.9578596+04:00

Spilling mistake in enrollment (double e)

```
-- 3)
select Course.title,
count(enrollment.trainee_id) as total_trainees
from schedule
inner join course on schedule.course_id = course.course_id
inner join enrollment on course.course_id = enrollment.course_id
```

1 %

Messages

Msg 8120, Level 16, State 1, Line 182
Column 'course.title' is invalid in the select list because it is not contained in either an aggr

Completion time: 2025-06-28T12:57:44.2179462+04:00

I discovered that the error happened because I used the column course.title in the select statement along with **group by** but I forgot to include that column in the group by clause. The solution was to add course.title along with the other columns in the group by so that SQL accepts the query without errors

```
-- 5)
select trainee_id, count(*) as total_courses
from Enrollment
group by trainee.name;

-- 6)
```

% ▼ ◄

Messages

Msg 207, Level 16, State 1, Line 155
Invalid column name 'name'.

Completion time: 2025-06-28T14:59:19.9101804+04:00

I realized the problem happened because I tried to use the column trainee.name directly in the query, but that column doesn't actually exist in the enrollment table — it only exists in the trainee table. Since I didn't write a join to connect both tables, SQL gave me an error saying the column is invalid. To fix it in a simpler way and keep the query short, I decided to use trainee_id instead because it already exists in the enrollment table, so I didn't need a join and the query stayed simple and clear.

```
-- 1)
select trainer_id, .title as cousrse_titile
from Schedule
inner join Course on Schedule.course_id = Course.course_id



-- 2)
select Trainer.name, Schedule.start_date, Schedule.end_date, Schedule.time_slot
from Schedule
```

l %   ▼ ◀

Messages

Msg 4104, Level 16, State 1, Line 168
The multi-part identifier ".title" could not be bound.

Completion time: 2025-06-28T15:13:44.0879330+04:00

I noticed the problem was that in the original query, there was a dot . mistakenly placed before the column name title, and the alias was misspelled as cousrse_titile. Also, the comma after trainer_id was missing. These small syntax errors caused the query to fail. To fix it, I removed the extra dot, corrected the alias to course_title, and added the missing comma. This made the SQL query valid and clear.

Name: Maha Al Yahmadi

Batch: Data Science