

SCHOOL MANAGEMENT SYSTEM

Presented By: Darina Burdyaeva

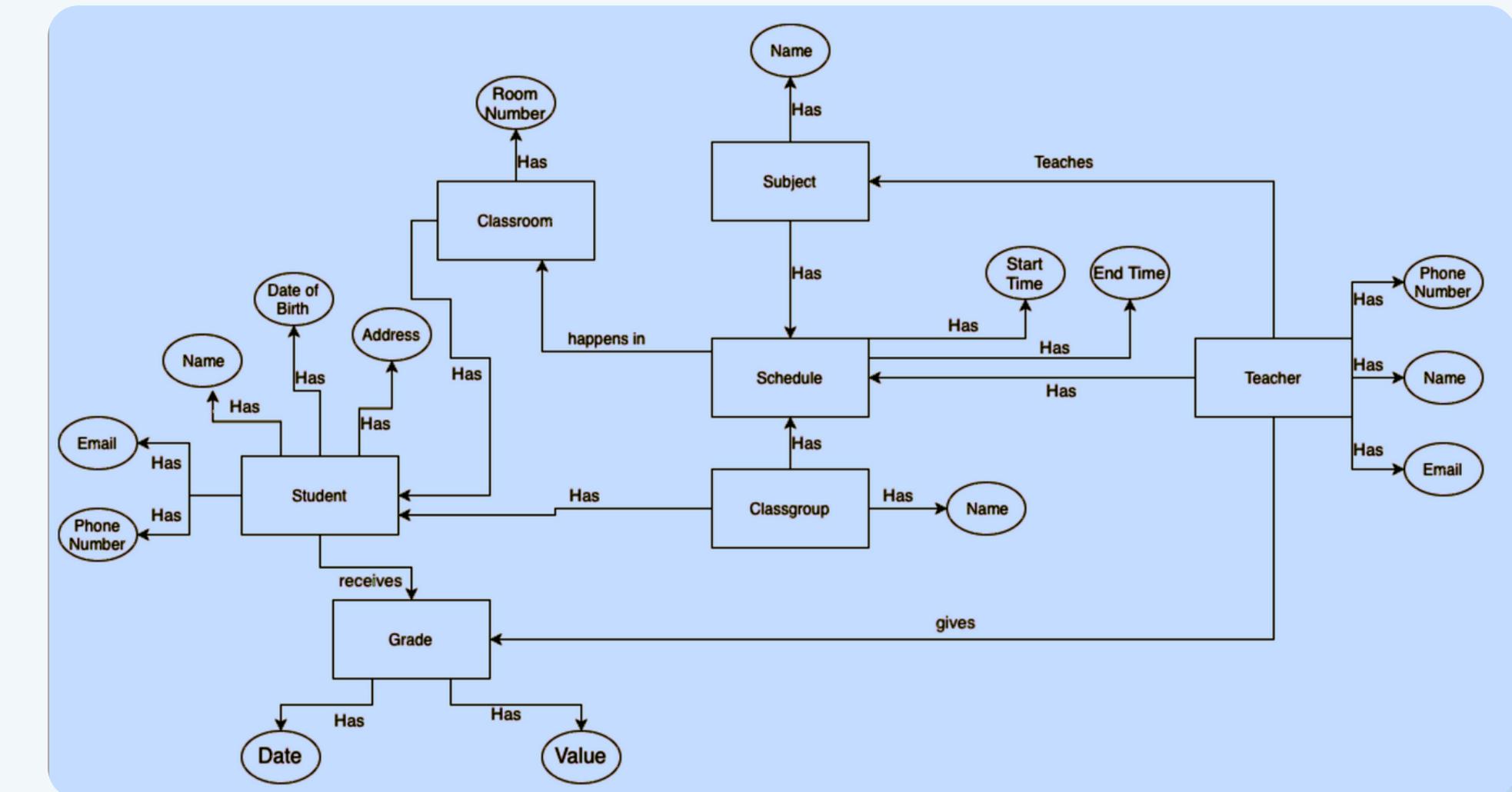


DataBase

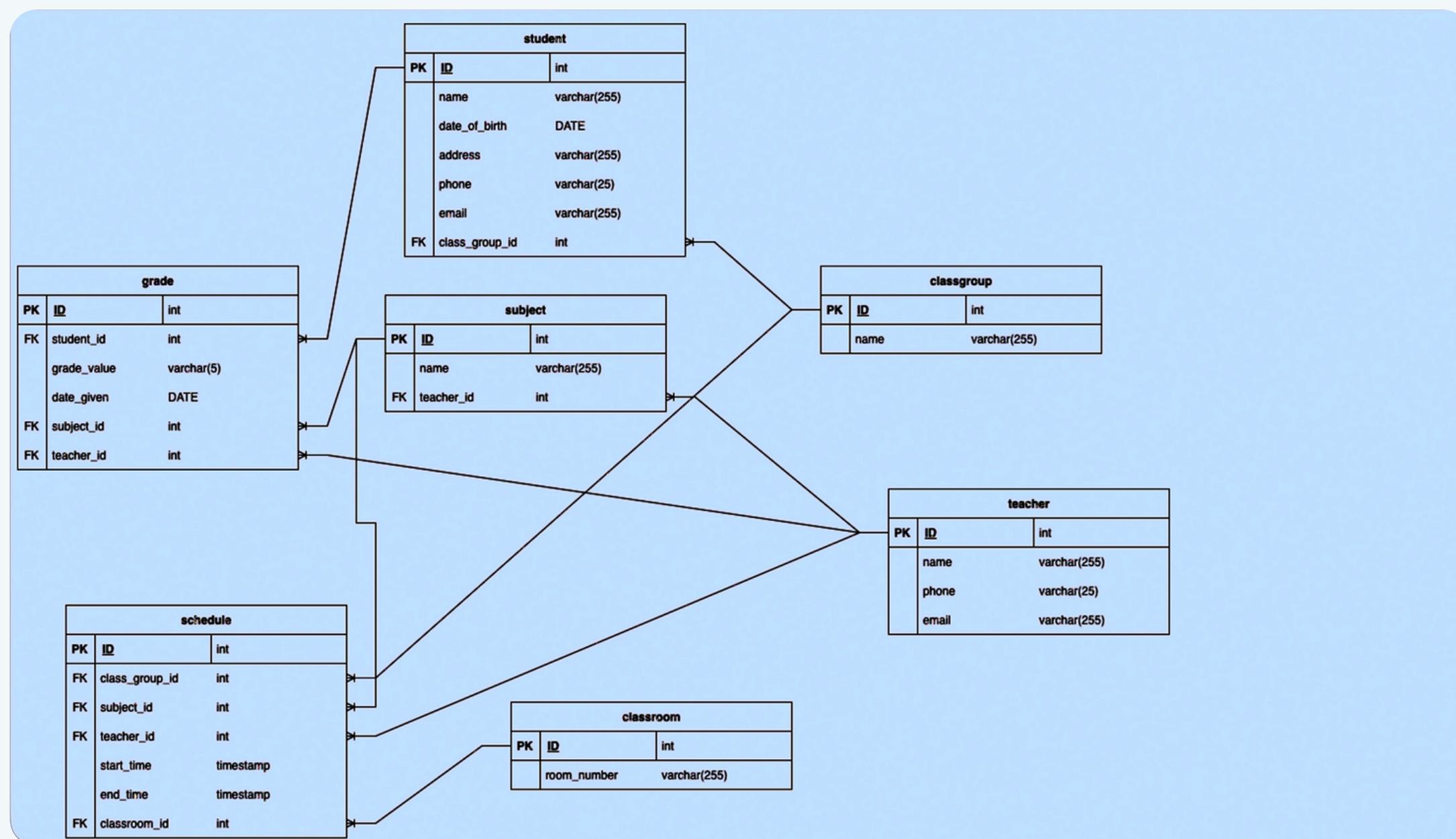
Final Project

2025

CONCEPTUAL DIAGRAM



LOGICAL DIAGRAM



O2

BASIC SQL QUERIES

01

```
SELECT * FROM Student;
```

select all students:

02

```
SELECT * FROM Student  
WHERE class_group_id = 1;
```

students from one class group

03

```
SELECT name, phone FROM Teacher;
```

names of all teachers and their mobile phone

04

```
SELECT  
    s.name AS student_name,  
    cg.name AS class_group  
FROM Student s  
JOIN ClassGroup cg ON  
cg.id = s.class_group_id;
```

student names with their class group

ADVANCED SQL QUERIES

01

full schedule with subject, teacher, class group and classroom names

```
SELECT
    sch.id,
    sub.title AS subject,
    t.name AS teacher,
    cg.name AS class_group,
    cr.room_number AS classroom,
    sch.start_time,
    sch.end_time
FROM Schedule sch
    JOIN Subject sub ON sub.id = sch.subject_id
    JOIN Teacher t ON t.id = sch.teacher_id
    JOIN ClassGroup cg ON cg.id = sch.class_group_id
    JOIN Classroom cr ON cr.id = sch.classroom_id
ORDER BY sch.start_time;
```

02

average grade per subject

```
SELECT
    sub.title AS subject,
    ROUND(
        AVG(
            CASE
                WHEN grade_value = 'A' THEN 5
                WHEN grade_value = 'B' THEN 4
                WHEN grade_value = 'C' THEN 3
                WHEN grade_value = 'D' THEN 2
                ELSE 1
            END
        ), 1) AS avg_grade_numeric
FROM Grade g
    JOIN Subject sub ON sub.id = g.subject_id
GROUP BY sub.title
ORDER BY avg_grade_numeric DESC;
```

ADVANCED SQL QUERIES

03 count students in each class group

```
SELECT
    cg.name AS class_group,
    COUNT(s.id) AS student_count
FROM ClassGroup cg
    LEFT JOIN Student s ON s.class_group_id = cg.id
GROUP BY cg.name
ORDER BY student_count DESC;
```

04 grades with student + subject + teacher names

```
SELECT
    g.id,
    s.name AS student,
    sub.title AS subject,
    t.name AS teacher,
    g.grade_value,
    g.date_given
FROM Grade g
    JOIN Student s ON s.id = g.student_id
    JOIN Subject sub ON sub.id = g.subject_id
    JOIN Teacher t ON t.id = g.teacher_id
ORDER BY g.date_given DESC;
```

DataBase

Final Project

05

TRANSACTION

1. Generating the next available student ID.
2. Inserting the new student into the student table.
3. Inserting a grade for this student into the grade table.
4. Committing both operations together.

```
BEGIN;
WITH next_id AS (
    SELECT COALESCE(MAX(id), 0) + 1 AS new_id
    FROM Student
),
insert_student AS (
INSERT INTO Student (
    id,
    name,
    date_of_birth,
    address,
    class_group_id,
    email,
    phone
)
SELECT
    new_id,
    'Ban BoBin',
    '2005-01-01',
    'His Address 123',
    1,
    'test_bobin@example.com',
    '+996700000000'
FROM next_id
    | | | RETURNING id
)
INSERT INTO Grade (
    id,
    student_id,
    subject_id,
    teacher_id,
    grade_value,
    date_given
)
SELECT
    (SELECT COALESCE(MAX(id), 0) + 1 FROM Grade),
    id,
    3,
    2,
    'B',
    CURRENT_DATE
FROM insert_student;
COMMIT;
SELECT * FROM Student ORDER BY id DESC LIMIT 5;
SELECT * FROM Grade ORDER BY id DESC LIMIT 5;
```

THANK YOU

