



CHAPTER 6

TRUY VẤN DỮ LIỆU

Truy vấn dữ liệu Database Education

6.1. THỰC HIỆN 11 TRUY VẤN DỮ LIỆU

YÊU CẦU TRUY VẤN:

1. Cho biết ID, Tên học sinh trong thành phố có ID chứa số 6 và có họ là NGUYỄN
2. Cho biết ID, Tên giáo viên, đơn vị công tác và môn học giảng dạy của giáo viên trong thành phố
3. Liệt kê ra tất cả học sinh có học phí cao hơn mức học phí trung bình
4. Cho biết tất cả giáo viên trong thành phố đã đến độ tuổi nghỉ hưu hay chưa
5. Cho biết thông tin tất cả các trường THPT tại Quận Hai Bà Trưng
6. Cho biết danh sách tất cả giáo viên dạy 2 môn trở lên
7. Cho biết danh sách giáo viên có độ tuổi từ 20 đến 30 tuổi của trường THPT Ba Vì
8. Cho biết 10 học sinh nghỉ học nhiều nhất trong tất cả các trường
9. Cho biết danh sách học sinh còn thiếu thông tin của phụ huynh như Số điện thoại hoặc Email
10. Cập nhật bảng dữ liệu học sinh cho toàn thành phố Hà Nội
11. Tính điểm trung bình các môn cuối kì của học sinh toàn thành phố

6.1. THỰC HIỆN 11 TRUY VẤN DỮ LIỆU

SỬ DỤNG ĐA DẠNG CÁC CÂU LỆNH:

- SELECT, WHERE
- SELECT DISTINCT
- WITH
- JOIN (INNER JOIN, LEFT JOIN, CROSS JOIN)
- ALIAS, ORDER BY
- ON, AND, OR
- AVG(), COUNT(), MAX()
- CASE
- EXISTS, UNION
- GROUP BY
- HAVING
- BETWEEN AND
- LIMIT, LIKE
- HAVING
- NULL, INSERT INTO
- SUBQUERY

1. Cho biết ID, Tên học sinh trong thành phố có ID chứa số 6 và có họ là NGUYỄN:

1. Π student_id, student_name (σ student_id='%6%' ^ student_name='NGUYỄN%')(students)

```
1  -- #1
2  -- Cho biết ID, tên của học sinh có ID chứa số 6 và có họ là NGUYỄN
3  •  SELECT student_id, student_name
4     FROM students
5     WHERE student_id LIKE '%6%' AND student_name LIKE 'NGUYỄN%';
```

Result Grid		Filter Rows:
	student_id	student_name
▶	16	Nguyễn Thị Ngân Hà
	60	Nguyễn Phú Hải Phong
	64	Nguyễn Trường Thịnh
	66	Nguyễn Minh Thư
	76	Nguyễn Thùy Anh
	166	Nguyễn Tuấn Hưng
	168	Nguyễn Tiến Hưng
	196	Nguyễn Vũ Minh Đức
	256	Nguyễn Đăng Phong
	262	Nguyễn Hữu Thành
	264	Nguyễn Quang Thiện
	268	Nguyễn Hoàng Vy
	296	Nguyễn Huyền Thư
	306	Nguyễn Phan Anh
	316	Nguyễn Minh Hà
	326	Nguyễn Phương Linh
	360	Nguyễn Thảo Linh
	362	Nguyễn Trà My
	367	Nguyễn Phương Khán...

2. Cho biết ID, Tên giáo viên, đơn vị công tác và môn học giảng dạy của giáo viên trong thành phố:

```
1 WITH infor_teacher as(  
2     SELECT DISTINCT teacher_id, teacher_name,  
3         (SELECT school_name FROM schools AS s INNER JOIN teachers AS t ON s.school_id = t.school_id)  
4         AS 'school',  
5         (SELECT subject_name FROM subject AS s INNER JOIN teachers AS t ON s.subject_id = t.subject_id )  
6         AS 'subject'  
7     FROM teachers AS t INNER JOIN schools AS s ON t.school_id = s.school_id  
8     INNER JOIN subject AS sb ON t.subject_id = sb.subject_id  
9 )  
10 SELECT teacher_id, teacher_name,  
11 'school' AS school_name, 'subject' AS subject_name  
12 FROM infor_teacher  
13 ORDER BY teacher_id ASC
```

2. $R2 = \text{teachers} \bowtie \text{schools} \bowtie \text{subject}$

$\Pi_{\text{teacher_id, teacher_name, school_name, subject_name}}(R2)$

	teacher_id	teacher_name	school_name	subject_name
▶	1	Nguyễn Phú Đồng	THPT Minh Phú	Vật lí
	2	Huỳnh Văn Nhứt	THPT Xuân Thủy - Từ Liêm	Vật lí
	3	Nguyễn Thị Minh Hằng	THPT năng khiếu TDTT Hà Nội	Vật lí
	4	Trần Thị Thanh Hào	THPT Thống Nhất	Vật lí
	5	Lương Thị Phương	THPT Hermann Gmeiner	Vật lí
	6	Hồ Thị Quỳnh Giang	Trường THPT Thanh Oai A	Địa lí
	7	Đỗ Thị Ngọc Lan	THPT Đoàn Thị Điểm	Địa lí
	8	Nguyễn Thị Thanh Vân	THPT Chương Mỹ B	Địa lí
	9	Tạ Thị Hiệp	THPT Tân Lập	Địa lí
	10	Võ Thị Minh Thủy	THPT Lương Văn Can	GDQP-AN
	11	Nguyễn Việt Sơn	THPT Hà Đông	GDQP-AN
	12	Đỗ Thị Thanh Tiên	THPT Cầu Giấy	GDQP-AN
	13	Trần Văn Quang	THPT Nguyễn Du - Mê Linh	Hóa học
	14	Bùi Thanh Huyền	THPT Đông Mỹ	Hóa học
	15	Nguyễn Thị Xuân Hương	THPT Cao Bá Quát - Gia Lâm	Hóa học
	16	Đoàn Thị Dương	THPT năng khiếu TDTT Hà Nội	Hóa học
	17	Nguyễn Thuỳ Nữ Hiệp	THPT Minh Quang	Hóa học
	18	Lê Thị Thu Tâm	THPT Duyên Thái	Hóa học
	19	Phan Thị Điểm Quên	THPT Dương Xá	Hóa học

3. Liệt kê ra tất cả học sinh có học phí cao hơn mức học phí trung bình:

3. $R3 = \text{student} \bowtie \text{tuition_fee}$



$\prod_{\text{student_id, student_name, fee}} (\sigma_{\text{fee} > \text{hocphi.averange}}(R3))$

```
1  WITH hocPhi(average) AS(  
2      SELECT avg(fee) FROM tuition_fee  
3  )  
4  SELECT student_id, student_name, fee FROM students  
5  INNER JOIN tuition_fee ON students.id = tuition_fee.id  
6  JOIN hocPhi  
7  WHERE fee > hocPhi.average  
8  ORDER BY student_id
```

Result Grid				Filter Rows:
	student_id	student_name	fee	
▶	1	Đặng Tuấn Anh	860000	
	3	Lưu Trang Anh	980000	
	8	Trần Thị Minh Châu	940000	
	9	Tăng Phương Chi	830000	
	14	Mạc Trung Đức	850000	
	17	Nguyễn Lê Hiếu	860000	
	18	Phạm Xuân Hòa	860000	
	19	Khoa Minh Hoàng	920000	
	23	Trần Tuấn Hưng	810000	
	24	Phạm Gia Minh	810000	
	25	Đỗ Hoàng Mỹ	810000	
	29	Nguyễn Công Thành	900000	
	31	Nguyễn Hương Thảo	870000	
	32	Tô Diệu Thảo	970000	
	35	Đặng Thành Trung	1000000	
	37	Lê Khánh Vy	910000	

4. Cho biết tất cả giáo viên trong thành phố đã đến độ tuổi nghỉ hưu hay chưa:

```
1 • SELECT teacher_id, teacher_name, age,  
2 CASE  
3     WHEN t.age > 55 THEN 'Giáo viên đến độ tuổi nghỉ hưu'  
4     ELSE 'Giáo viên chưa đến độ tuổi nghỉ hưu'  
5 END AS age_retire  
6 FROM teachers AS t
```

Result Grid  Filter Rows: <input type="text"/> Export:  Wrap Cell Conte				
	teacher_id	teacher_name	age	age_retire
▶	1	Nguyễn Phú Đồng	56	Giáo viên đến độ tuổi nghỉ hưu
	2	Huỳnh Văn Nhứt	44	Giáo viên chưa đến độ tuổi nghỉ hưu
	3	Nguyễn Thị Minh Hằng	41	Giáo viên chưa đến độ tuổi nghỉ hưu
	4	Trần Thị Thanh Hào	35	Giáo viên chưa đến độ tuổi nghỉ hưu
	5	Lương Thị Phương	30	Giáo viên chưa đến độ tuổi nghỉ hưu
	6	Hồ Thị Quỳnh Giang	50	Giáo viên chưa đến độ tuổi nghỉ hưu
	7	Đỗ Thị Ngọc Lan	53	Giáo viên chưa đến độ tuổi nghỉ hưu
	8	Nguyễn Thị Thanh Vân	40	Giáo viên chưa đến độ tuổi nghỉ hưu
	9	Tạ Thị Hiệp	40	Giáo viên chưa đến độ tuổi nghỉ hưu
	10	Võ Thị Minh Thủy	30	Giáo viên chưa đến độ tuổi nghỉ hưu
	11	Nguyễn Việt Sơn	25	Giáo viên chưa đến độ tuổi nghỉ hưu
	12	Đỗ Thị Thanh Tiền	44	Giáo viên chưa đến độ tuổi nghỉ hưu
	13	Trần Văn Quang	58	Giáo viên đến độ tuổi nghỉ hưu
	14	Bùi Thanh Huyền	59	Giáo viên đến độ tuổi nghỉ hưu
	15	Nguyễn Thị Xuân Hương	57	Giáo viên đến độ tuổi nghỉ hưu

5. Cho biết thông tin tất cả các trường THPT tại Quận Hai Bà Trưng:

```
1 • SELECT school_name, emai FROM schools
2 WHERE EXISTS (SELECT*
3               FROM address
4               WHERE address.address_id = 3
5               AND schools.address_id = 3);
```



5. $R5 = \text{schools} \bowtie \text{address}$

$\Pi_{\text{school_name}, \text{email}} (\sigma_{\text{address_id}='3'}(R5))$

Result Grid			Filter Rows:	Export:	Wrap Cell Content:
	school_name	emai			
▶	TRƯỜNG THPT BA VỈ	http://thptbavi.edu.vn			
	TRƯỜNG THPT ĐOÀN KẾT - HAI BÀ TRƯNG	http://thptdoanket-hbt.edu.vn			
	THPT Nguyễn Văn Cừ	http://c3nguyenvancu.edu.vn			
	Trường Trung học phổ thông Lê Quý Đôn - Đống...	http://thptlequydon-dd.edu.vn			
	Trường THPT Hoàng Cầu	http://thpthoangcau.edu.vn			
	Trường THPT Xuân Mai	http://thptxuanmai.edu.vn			
	Trường THPT Thanh Oai A	http://thptthanhoaia.edu.vn			
	THPT Nguyễn Du Thanh Oai	http://nguyenduthanhoai.edu.vn			
	THPT Hồng Hà				
	THPT Hoàng Diệu				
	THPT Mai Hắc Đế				
	THPT Thăng Long				
	THPT Trần Nhân Tông				
	THCS - THPT Tạ Quang Bửu				

6. Cho biết danh sách tất cả giáo viên dạy 2 môn trở lên:

```
1  SELECT teacher_id, teacher_name, COUNT(subject_id) AS SoLuong
2  FROM teachers
3  GROUP BY teacher_name
4  HAVING COUNT(subject_id) > 2
5  ORDER BY teacher_id;
```



Result Grid   Filter Rows: <input type="text"/>			
	teacher_id	teacher_name	SoLuong
▶	1544	Thùy Oanh Hồ	3
	1642	Tùng Quang Vũ	3
	1708	Đức Toàn Đặng	3
	2464	Phú Hùng Phan	4
	2717	An Tường Mai	3
	2852	Quốc Hưng Đỗ	3
	2858	Xuân Hòa Đào	3
	4267	Trúc Lâm Mai	3

6. $\text{soluong} \leftarrow \text{COUNT}(\text{subject})$

$\Pi_{\text{teacher_id}, \text{teacher_name}, \text{soluong}}(\sigma_{\text{COUNT}(\text{subject_id}) > 2}(\text{teachers}))$

7. Cho biết danh sách giáo viên có độ tuổi từ 20 đến 30 tuổi của trường THPT Ba Vì:

```
1 • SELECT teacher_id, teacher_name, age, school_name
2 FROM teachers INNER JOIN schools
3 ON teachers.school_id = 1
4 AND schools.school_id = 1
5 WHERE age BETWEEN 20 AND 30
6 ORDER by teacher_id ASC
```

Result Grid  Filter Rows: <input type="text"/> Export: 				
	teacher_id	teacher_name	age	school_name
▶	995	Tuệ Mẫn Đặng	28	TRƯỜNG THPT BA VÌ
	5384	Khiết Tâm Tăng	26	TRƯỜNG THPT BA VÌ
	5643	Thụy Nương Tăng	28	TRƯỜNG THPT BA VÌ

7. $R_{73} = \text{teachers} \bowtie \text{schools}$


$R_{71} = \Pi_{\text{teacher_id}, \text{teacher_name}, \text{age}} (\sigma_{\text{age} > 20 \wedge \text{age} < 30}(\text{teachers}))$

$R_{72} = \Pi_{\text{teacher_id}, \text{school_name}} (\sigma_{\text{school_id} = '1'}(R_{73}))$

$R_7 = R_{71} \bowtie R_{72}$

8. Cho biết 10 học sinh nghỉ học nhiều nhất trong tất cả các trường:

```
1 • WITH absen(maxium) AS(  
2     SELECT MAX(num_of_absences)  
3         FROM attendance  
4 )  
5     SELECT student_id, student_name  
6         FROM students  
7     WHERE student_id IN (SELECT student_id  
8                         FROM attendance, absen  
9                         WHERE num_of_absences = absen.maxium)  
10    ORDER BY student_id ASC  
11    LIMIT 10;
```

Result Grid  Filter Rows: <input type="text"/>		
	student_id	student_name
▶	11	Phạm Tiến Dũng
	40	Nguyễn Quốc Huy
	142	Nguyễn Đoàn Đức Thành
	144	Đào Hồng Thiện
	146	Phạm Nguyễn Minh Trí
	152	Nguyễn Trần Bình
	198	Lưu Hương Giang
	229	Phạm Phương Anh
	292	Nguyễn Hoàng Thái
	340	Lương Phú An

9. Cho biết danh sách học sinh còn thiếu thông tin của phụ huynh như Số điện thoại hoặc Email:

```
1 • USE education;
2 • WITH id(wherenull) AS(
3     SELECT student_id FROM parents AS p
4         WHERE p.phone IS NULL
5             OR p.email IS NULL
6 )
7 SELECT student_id, student_name FROM students AS s, id
8     WHERE s.student_id = id.wherenull
9     ORDER BY s.student_id
```

Result Grid			Filter Rows:
	student_id	student_name	
▶	3	Lưu Trang Anh	
	5	Phạm Thị Hiền Anh	
	7	Đỗ Hoàng Gia Bảo	
	8	Trần Thị Minh Châu	
	11	Phạm Tiến Dũng	
	12	Nguyễn Thái Dương	
	13	Trần An Dương	
	14	Mạc Trung Đức	
	15	Vũ Hương Giang	
	16	Nguyễn Thị Ngân Hà	
	18	Phạm Xuân Hòa	
	19	Khoa Minh Hoàng	
	20	Nguyễn Hữu Hiệp H...	
	21	Nguyễn Mạnh Hùng	
	22	Nguyễn Vũ Gia Hưng	

9. $R9 = \text{students} \bowtie \text{parents}$

$\Pi_{\text{student_id}, \text{student_name}}(\sigma_{\text{phone}="" \vee \text{email}=""}(R9))$

9. Cho biết danh sách học sinh còn thiếu thông tin của phụ huynh như Số điện thoại hoặc Email:

```
1 • USE education;
2 • SELECT s.student_id, s.student_name, p.phone, p.email
3 FROM students AS s LEFT JOIN parents AS p
4 ON s.student_id = p.student_id
5 WHERE p.phone IS NULL
6      OR p.email IS NULL
7 ORDER BY student_id ASC
```

	student_id	student_name	phone	email
▶	3	Lưu Trang Anh	860827484	NULL
	5	Phạm Thị Hiền Anh	862474653	NULL
	7	Đỗ Hoàng Gia Bảo	868040308	NULL
	8	Trần Thị Minh Châu	860763479	NULL
	11	Phạm Tiến Dũng	862882269	NULL
	12	Nguyễn Thái Dương	869683336	NULL
	13	Trần An Dương	866675546	NULL
	14	Mạc Trung Đức	861179457	NULL
	15	Vũ Hương Giang	860087087	NULL
	16	Nguyễn Thị Ngân Hà	865256608	NULL
	18	Phạm Xuân Hòa	866066134	NULL
	19	Khoa Minh Hoàng	869698079	NULL
	20	Nguyễn Hữu Hiệp H...	866577442	NULL
	21	Nguyễn Mạnh Hùng	861764532	NULL
	22	Nguyễn Vũ Gia Hưng	861542004	NULL
	24	Phạm Gia Minh	860390262	NULL
	25	Đỗ Hoàng Mỹ	863074229	NULL
	27	Đàm Yến Nhi	863312244	NULL

9. $R9 = \text{students} \bowtie \text{parents}$

$\Pi_{\text{student_id}, \text{student_name}}(\sigma_{\text{phone}="" \vee \text{email}=""}(R9))$

10. Cập nhật bảng dữ liệu học sinh cho toàn thành phố Hà Nội

```
INSERT INTO `new_education`.`students_1`
(`student_id`, `student_name`, `date_of_birth`, `gender`, `id`)
VALUES (1, 'Đặng Tuấn Anh', '2006/1/17', 'Nam', 1565), (2, 'Hoàng Đức Anh', '2006/5/8', 'Nam', 2224),
(3, 'Lưu Trang Anh', '2006/3/15', 'Nữ', 2099), (4, 'Phạm Hoàng Anh', '2006/7/16', 'Nam', 1973),
(5, 'Phạm Thị Hiền Anh', '2006/2/11', 'Nữ', 1843), (6, 'Phạm Khắc Việt Anh', '2006/11/27', 'Nam', 876),
(7, 'Đỗ Hoàng Gia Bảo', '2006/12/3', 'Nam', 311), (8, 'Trần Thị Minh Châu', '2006/10/10', 'Nữ', 818),
(9, 'Tăng Phương Chi', '2006/2/15', 'Nữ', 2127), (10, 'Gan Feng Du', '2006/8/9', 'Nữ', 1918),
(11, 'Phạm Tiến Dũng', '2006/9/15', 'Nam', 1226), (12, 'Nguyễn Thái Dương', '2006/2/24', 'Nam', 1584),
(13, 'Trần An Dương', '2006/8/11', 'Nam', 378), (14, 'Mạc Trung Đức', '2006/4/3', 'Nam', 1630),
(15, 'Vũ Hương Giang', '2006/4/23', 'Nữ', 727), (16, 'Nguyễn Thị Ngân Hà', '2006/12/21', 'Nữ', 329),
(17, 'Nguyễn Lê Hiếu', '2006/7/3', 'Nam', 1169), (18, 'Phạm Xuân Hòa', '2006/1/1', 'Nam', 1565),
(19, 'Khoa Minh Hoàng', '2006/1/19', 'Nam', 1767), (20, 'Nguyễn Hữu Hiệp Hoàng', '2006/11/18', 'Nam', 988),
(21, 'Nguyễn Mạnh Hùng', '2006/10/22', 'Nam', 535), (22, 'Nguyễn Vũ Gia Hưng', '2006/1/18', 'Nam', 1160),
(23, 'Trần Tuấn Hưng', '2006/6/8', 'Nam', 807), (24, 'Phạm Gia Minh', '2006/6/23', 'Nam', 882),
(25, 'Đỗ Hoàng Mỹ', '2006/11/4', 'Nữ', 2146), (26, 'Đỗ Quang Ngọc', '2006/8/8', 'Nam', 2014),
(27, 'Đàm Văn Nhi', '2006/6/24', 'Nữ', 2246), (28, 'Đào Hoàng Sơn', '2006/12/7', 'Nam', 1584)
```

	student_id	student_name	date_of_birth	gender	id
▶	1	Đặng Tuấn Anh	2006/1/17	Nam	1565
	2	Hoàng Đức Anh	2006/5/8	Nam	2224
	3	Lưu Trang Anh	2006/3/15	Nữ	2099
	4	Phạm Hoàng Anh	2006/7/16	Nam	1973
	5	Phạm Thị Hiền Anh	2006/2/11	Nữ	1843
	6	Phạm Khắc Việt Anh	2006/11/27	Nam	876
	7	Đỗ Hoàng Gia Bảo	2006/12/3	Nam	311
	8	Trần Thị Minh Châu	2006/10/10	Nữ	818
	9	Tăng Phương Chi	2006/2/15	Nữ	2127
	10	Gan Feng Du	2006/8/9	Nữ	1918
	11	Phạm Tiến Dũng	2006/9/15	Nam	1226
	12	Nguyễn Thái Dương	2006/2/24	Nam	1584
	13	Trần An Dương	2006/8/11	Nam	378
	14	Mạc Trung Đức	2006/4/3	Nam	1630
	15	Vũ Hương Giang	2006/4/23	Nữ	727

```
INSERT INTO `new_education`.`students_1`
(`student_id`, `student_name`, `date_of_birth`, `gender`, `id`)
VALUES (1, 'Đặng Tuấn Anh', '2006/1/17', 'Nam', 1565)
```


11. Tính điểm trung bình các môn cuối kì của học sinh toàn thành phố:

```
1  • WITH final(scor) AS(  
2    SELECT final_score FROM scores  
3    UNION  
4    SELECT final_score FROM scores_2  
5    UNION  
6    SELECT final_score FROM scores_3  
7    UNION  
8    SELECT final_score FROM scores_4  
9  )  
10 SELECT AVG(final.scor) AS avg_final FROM final
```

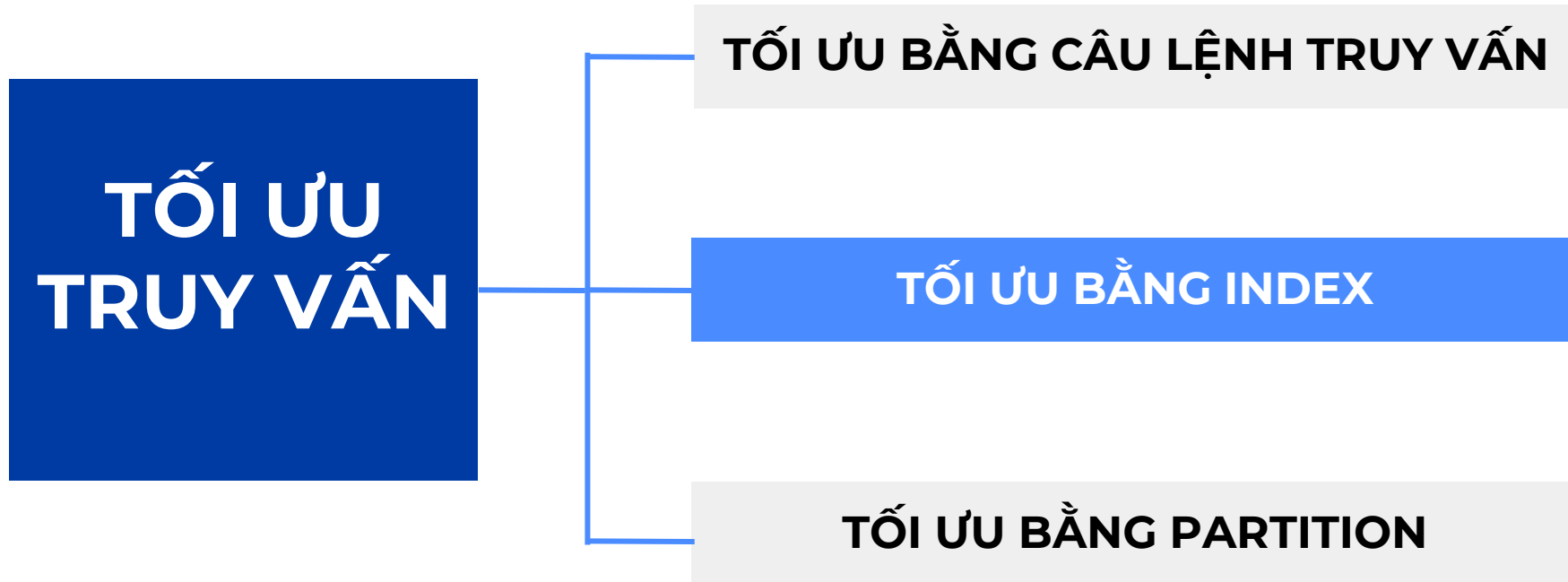
Result Grid	
	avg_final
▶	5.0000



6.2 TỐI ƯU TRUY VẤN

Thực hiện các công việc để tối ưu truy vấn

6.2.TỐI ƯU TRUY VẤN



TỐI ƯU BẢNG CÂU LỆNH TRUY VẤN

- Chỉ lấy ra những cột cần thiết thay vì `SELECT * FROM`
- Tránh sử dụng function đối với cột và bảng
- Tránh các ký tự %, -,... ở đầu giá trị tìm kiếm `LIKE`
- Dùng `INNER JOIN` thay vì dùng `SUBQUERY`

TỐI ƯU BẰNG CÂU LỆNH TRUY VẤN

- ```
SELECT student_id, student_name FROM students
WHERE student_id
IN (SELECT student_id FROM parents
 WHERE phone LIKE '8%')
```

Duration / Fetch  
0.000 sec / 0.203 sec

- ```
SELECT student_id, student_name
FROM students INNER JOIN parents
USING (student_id)
WHERE phone LIKE '8%'
```

Duration / Fetch
0.000 sec / 0.078 sec

Result Grid		Filter Rows:
	student_id	student_name
	1	Đặng Tuấn Anh
	2	Hoàng Đức Anh
	3	Lưu Trang Anh
	4	Phạm Hoàng Anh
	5	Phạm Thị Hiền Anh
	6	Phạm Khắc Việt Anh
	7	Đỗ Hoàng Gia Bảo
	8	Trần Thị Minh Châu
	9	Tăng Phương Chi
	10	Gan Feng Du
	11	Phạm Tiến Dũng
	12	Nguyễn Thái Dương
	13	Tiến An Dương

TỐI ƯU BẰNG INDEX

KHÔNG SỬ DỤNG INDEX

```
SELECT student_id, subject_id, oral_score, midterm_score, final_score  
FROM scores  
WHERE final_score >= 7
```

00:00:07 | 1,300,139 rows

	student_id	subject_id	oral_score	midterm_score	final_score
1	209100	8	6	8	7
2	209103	8	7	8	7
3	209105	8	4	9	7
4	209114	8	3	6	7
5	209115	8	7	9	7
6	209118	8	8	7	7
7	209129	8	4	5	7
8	210952	8	8	7	7
9	210961	8	7	9	7
10	210968	8	7	6	7
11	210991	8	6	8	7
12	210999	8	7	9	7
13	211003	8	1	6	7
14	211016	8	4	5	7
15	211025	8	7	9	7
16	211035	8	1	4	7
17	211051	8	2	5	7

Query executed successfully. TUANHUNG\TUANHUNG (16.0 RTM) | sa (66) | EDUCATION | 00:00:07 | 1,300,139 rows

SỬ DỤNG INDEX

```
CREATE CLUSTERED INDEX index_scores_final ON scores(final_score)
```

```
SELECT student_id, subject_id, oral_score, midterm_score, final_score  
FROM scores  
WHERE final_score >= 7
```

00:00:05 | 1,300,139 rows

	student_id	subject_id	oral_score	midterm_score	final_score
1	209100	8	6	8	7
2	209103	8	7	8	7
3	209105	8	4	9	7
4	209114	8	3	6	7
5	209115	8	7	9	7
6	209118	8	8	7	7
7	209129	8	4	5	7
8	210952	8	8	7	7
9	210961	8	7	9	7
10	210968	8	7	6	7
11	210991	8	6	8	7
12	210999	8	7	9	7
13	211003	8	1	6	7
14	211016	8	4	5	7
15	211025	8	7	9	7
16	211035	8	1	4	7
17	211051	8	2	5	7

Query executed successfully. TUANHUNG\TUANHUNG (16.0 RTM) | sa (66) | EDUCATION | 00:00:05 | 1,300,139 rows

TỐI ƯU BẰNG PARTITION

KHÔNG SỬ DỤNG PARTITION

SQLQuery4.sql - T...DUCATION (sa (51))* SQLQuery3.sql - T...DUCATION (sa (65))*

```
SELECT * FROM scores WHERE final_score > 6
```

200 %

Results Messages

	student_id	subject_id	oral_score	midterm_score	final_score
1	1	12	2	7	8
2	1	5	2	3	8
3	1	8	2	6	8
4	1	4	7	3	8
5	1	4	7	3	8
6	1	10	9	6	9
7	1	3	7	3	7
8	2	12	8	7	7
9	2	7	7	8	9
10	2	5	1	2	7
11	2	11	3	7	9
12	3	7	3	5	9
13	3	4	4	9	9
14	3	5	5	9	7
15	3	4	4	9	9
16	3	11	9	8	7
17	3	10	8	7	9
18	4	10	2	7	7
19	4	6	2	4	8
20	4	7	1	3	9
21	4	6	2	4	8
22	4	12	6	4	9
23	5	5	1	7	7
24	5	4	2	5	8
25	5	4	2	5	8
26	5	2	1	8	8
27	6	1	7	3	7
28	6	8	5	2	9
29	6	4	4	5	9

00:00:05 | 1,300,139 rows

Query executed successfully. TUANHUNG\TUANHUNG (16.0 RTM) sa (51) EDUCATION 00:00:05 1,300,139 rows

SỬ DỤNG PARTITION

SQLQuery4.sql - T...DUCATION (sa (51))* SQLQuery3.sql - T...DUCATION (sa (65))*

```
CREATE PARTITION FUNCTION ScorePartitionFunction (float)
AS RANGE RIGHT FOR VALUES (6);

CREATE PARTITION SCHEME ScorePartitionScheme
AS PARTITION ScorePartitionFunction
TO ([file1], [file2]);

CREATE CLUSTERED INDEX CIX_scores ON scores(student_id)
WITH (DROP_EXISTING = ON)
ON ScorePartitionScheme(student_id);

SELECT *
FROM scores
WHERE final_score > 6;
```

150 %

Results Messages

	student_id	subject_id	oral_score	midterm_score	final_score
1	9045	12	2	9	7
2	9046	5	9	6	9
3	9046	2	4	5	7
4	9046	8	5	7	8
5	9046	7	4	3	8
6	9047	10	4	6	8
7	9047	4	7	4	7
8	9047	5	9	6	7
9	9047	4	7	4	7
10	9048	5	5	2	9
11	9048	7	3	5	9
12	9048	11	8	8	8
13	9048	1	8	3	8

00:00:04 | 1,300,139 rows

Query executed successfully. TUANHUNG\TUANHUNG (16.0 RTM) sa (65) EDUCATION 00:00:04 1,300,139 rows

CHAPTER 7

CẬP NHẬT DỮ LIỆU



[-] Programmability		
[-] Stored Procedures		
+ System Stored Procedures		
+ dbo.delete_address	+ [] dbo.select_address	+ [] dbo.copy_address
+ dbo.delete_attendance	+ [] dbo.select_attendance	+ [] dbo.copy_attendance
+ dbo.delete_class	+ [] dbo.select_class	+ [] dbo.copy_class_school
+ dbo.delete_class_school	+ [] dbo.select_class_school	+ [] dbo.copy_classes
+ dbo.delete_parents	+ [] dbo.select_parents	+ [] dbo.copy_parents
+ dbo.delete_schools	+ [] dbo.select_schools	+ [] dbo.copy_schools
+ dbo.delete_scores	+ [] dbo.select_scores	+ [] dbo.copy_scores
+ dbo.delete_students	+ [] dbo.select_students	+ [] dbo.copy_students
+ dbo.delete_subject	+ [] dbo.select_subject	+ [] dbo.copy_subject
+ dbo.delete_teachers	+ [] dbo.select_teachers	+ [] dbo.copy_teachers
+ dbo.delete_tuition_fee	+ [] dbo.select_tuition_fee	+ [] dbo.copy_tuition_fee
+ dbo.insert_address	+ [] dbo.update_address	
+ dbo.insert_attendance	+ [] dbo.update_attendance	
+ dbo.insert_class	+ [] dbo.update_class	
+ dbo.insert_class_school	+ [] dbo.update_class_school	
+ dbo.insert_parents	+ [] dbo.update_parents	
+ dbo.insert_schools	+ [] dbo.update_schools	
+ dbo.insert_scores	+ [] dbo.update_scores	
+ dbo.insert_students	+ [] dbo.update_students	
+ dbo.insert_subject	+ [] dbo.update_subject	
+ dbo.insert_teachers	+ [] dbo.update_teachers	
+ dbo.insert_tuition_fee	+ [] dbo.update_tuition_fee	

55 PROCEDURE

- 11 PROCEDURE UPDATE
- 11 PROCEDURE SELECT
- 11 PROCEDURE INSERT
- 11 PROCEDURE DELETE
- 11 PROCEDURE COPY

7.1.PROCEDURE SELECT ADDRESS:

```
SET ANSI_NULLS ON
```

```
GO
```

```
SET QUOTED_IDENTIFIER ON
```

```
GO
```

```
CREATE PROCEDURE select_address
```

```
AS
```

```
BEGIN
```

```
    SET NOCOUNT ON;
```

```
    SELECT * FROM address order by address_id
```

```
END
```

```
GO
```

SQLQuery1.sql - T...DUCATION (sa (66))*

select_address

200 %

Results Messages

	address_id	address_name
1	1	Ba Đình
2	2	Đồng Đa
3	3	Hai Bà Trưng
4	4	Hoàn Kiếm
5	5	Tây Hồ
6	6	Thanh Xuân
7	7	Hoàng Mai
8	8	Long Biên
9	9	Cầu Giấy
10	10	Bắc Từ Liêm
11	11	Nam Từ Liêm
12	12	Hà Đông
13	13	Thanh Trì
14	14	Đống Anh
15	15	Thường Tín
16	16	Mê Linh
17	17	Sóc Sơn
18	18	Thạch Thất
19	19	Quốc Oai
20	20	Thanh Oai
21	21	Đan Phượng
22	22	Gia Lâm
23	23	Chương Mỹ
24	24	Hoài Đức
25	25	Ba Vì
26	26	Mỹ Đức
27	27	Phủ Xuyên
28	28	Ứng Hòa
29	29	Sơn Tây
30	30	Phúc Thọ

Query executed successfully.

7.1.PROCEDURE INSERT ADDRESS:

```
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO

CREATE PROCEDURE insert_address(
@address_id float,
@address_name nvarchar(255))

AS
BEGIN
    insert into address (address_id, address_name)
    VALUES (@address_id, @address_name)
END
```

SQLQuery5.sql - T...DUCATION (sa (54))* x SQLQuery4.sql - T...DUCATION (sa (76))*

```
insert_address @address_id = 31, @address_name = 'Ha Noi'
select_address
```

200 %

Results Messages

	address_id	address_name
9	9	Cầu Giấy
10	10	Bắc Từ Liêm
11	11	Nam Từ Liêm
12	12	Hà Đông
13	13	Thanh Trì
14	14	Đống Anh
15	15	Thường Tín
16	16	Mê Linh
17	17	Sóc Sơn
18	18	Thạch Thất
19	19	Quốc Oai
20	20	Thanh Oai
21	21	Đan Phượng
22	22	Gia Lâm
23	23	Chương Mỹ
24	24	Hoài Đức
25	25	Ba Vì
26	26	Mỹ Đức
27	27	Phủ Xuyên
28	28	Ứng Hòa
29	29	Sơn Tây
30	30	Phúc Thọ
31	31	Ha Noi

Query executed successfully. TUANHUNG\TUANHUNG (16.0 R

7.1.PROCEDURE UPDATE ADDRESS:

```
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE update_address(
@address_id float,
@address_name nvarchar(255))
AS
BEGIN
    UPDATE address
    SET address_name = @address_name
    WHERE address_id = @address_id
END
```

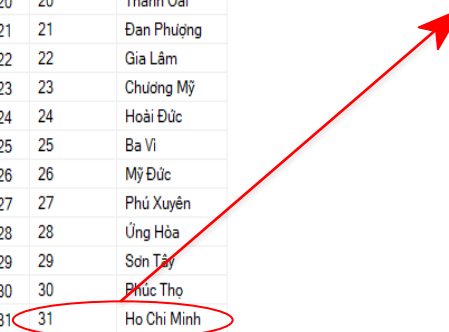
SQLQuery7.sql - T...DUCATION (sa (53)) SQLQuery6.sql - T...DUCATION (sa (68))* SQLQuery5.sql - T...DUCATION (sa (54))* ✖

```
update_address @address_id = 31, @address_name = 'Ho Chi Minh'
select_address
```

200 %

Results Messages

	address_id	address_name
9	9	Cầu Giấy
10	10	Bắc Từ Liêm
11	11	Nam Từ Liêm
12	12	Hà Đông
13	13	Thanh Trì
14	14	Đống Anh
15	15	Thường Tín
16	16	Mê Linh
17	17	Sóc Sơn
18	18	Thạch Thất
19	19	Quốc Oai
20	20	Thanh Oai
21	21	Đan Phượng
22	22	Gia Lâm
23	23	Chương Mỹ
24	24	Hoài Đức
25	25	Ba Vì
26	26	Mỹ Đức
27	27	Phủ Xuyên
28	28	Ứng Hòa
29	29	Sơn Tây
30	30	Phúc Thọ
31	31	Ho Chi Minh



31	31	Ho Chi Minh
----	----	-------------

Query executed successfully. TUANHUNG\TUANHUNG (16.0 RTM) sa (54)

7.1.PROCEDURE DELETE ADDRESS:

```
SET ANSI_NULLS ON
```

```
GO
```

```
SET QUOTED_IDENTIFIER ON
```

```
GO
```

```
CREATE PROCEDURE delete_address(  
@address_id float)
```

```
AS
```

```
BEGIN
```

```
DELETE FROM address
```

```
WHERE address_id = @address_id
```

```
END
```

SQLQuery9.sql - T...DUCATION (sa (74))* SQLQuery5.sql - T...DUCATION (sa (54))*

```
delete_address @address_id = 31  
select_address
```

200 %

Results Messages

	address_id	address_name
8	8	Long Biên
9	9	Cầu Giấy
10	10	Bắc Từ Liêm
11	11	Nam Từ Liêm
12	12	Hà Đông
13	13	Thanh Trì
14	14	Đống Anh
15	15	Thường Tín
16	16	Mê Linh
17	17	Sóc Sơn
18	18	Thạch Thất
19	19	Quốc Oai
20	20	Thanh Oai
21	21	Đan Phượng
22	22	Gia Lâm
23	23	Chương Mỹ
24	24	Hoài Đức
25	25	Ba Vì
26	26	Mỹ Đức
27	27	Phủ Xuyên
28	28	Ứng Hòa
29	29	Sơn Tây
30	30	Phúc Thọ

30 30 Phúc Thọ

Query executed successfully.

7.1.PROCEDURE INSERT SUBJECTS

```
1  DELIMITER $$
2
3  CREATE PROCEDURE INSERT_COURSE(
4      IN p_id VARCHAR(50),
5      IN p_name VARCHAR(20))
6
7  BEGIN
8
9      START TRANSACTION;
10
11     SELECT @subject_id:=MAX(subject_id) + 1 FROM subject;
12
13     INSERT INTO subject(subject_id, subject_name)
14     VALUES(@subject_id, p_id, p_name);
15
16     COMMIT;
17
18 END$$
19
20 DELIMITER ;
```

7.1. CREATE TRIGGER

```
1  --- Create Trigger insert transcript
2  CREATE TRIGGER trginsert_Diem
3  ON scores
4  For insert
5  as
6  DECLARE @oral_score int
7  DECLARE @midterm_score int
8  DECLARE @final_score int
9  SELECT @oral_score=oral_score,@midterm_score=midtermscore,@final_score
10 FROM insert_scores
11 IF (@oral_score <0) or (@oral_score>10) or (@midterm_score <0)
12 or (@midterm_score >10) or (@final_score < 0) or (@final_score > 10)
13 begin
14 print 'BAN DA NHAP SAI DIEM'
15 Rollback transaction
16 End
17 else
18 begin
19 Print 'NHAP DIEM THANH CONG'
20 End
```

KHÔNG CHO PHÉP NHẬP VÀO BẢNG ĐIỂM 1
GIÁ TRỊ ĐIỂM KHÔNG HỢP LỆ



7.2

TRANSACTION

Thao tác dữ liệu từ nghiệp vụ thực tế sử dụng TRANSACTION

THỰC HIỆN 5 NGHIỆP VỤ THỰC TẾ

1. Thêm một học sinh mới vào CSDL
2. Thêm một môn học vào chương trình đào tạo
3. Xóa một học sinh ra khỏi CSDL
4. Sửa thông tin học sinh trong CSDL
5. Cập nhật điểm môn học chi tiết cho một học sinh

1. Thêm một học sinh mới vào CSDL:

- Học sinh A có ngày sinh là “birthday”, giới tính “gender_student” và có mã trường lớp là id_student
- Mã học sinh ID_new_student = MAX(student_id) trong bảng students

```
1 • START TRANSACTION;  
2  
3 • SELECT ID_new_student = MAX(ID) + 1 FROM students;  
4  
5 • INSERT INTO students(student_id, student_name, date_of_birth, gender, id)  
6     VALUES(ID_new_student, "A", birth, gender, id_student);  
7  
8 • COMMIT;
```


2. Thêm một môn học vào chương trình đào tạo:

Các bước thực hiện

- Bắt đầu Transaction
- Lấy mã môn học $\text{new_subject_id} = \text{MAX}(\text{subject_id}) + 1$ từ bảng subject
- Thêm môn học mới với mã môn học là `new_subject_id` vào bảng subject
- Cập nhật vào bảng teachers thông tin giáo viên giảng dạy môn học mới với id là `new_subject_id`
- Cập nhật vào bảng scores điểm thi của môn học mới với môn học mới có id là `new_subject_id`
- Thực hiện Commit

2. Thêm một môn học vào chương trình đào tạo:

Cài đặt:

```
1 • START TRANSACTION;
2
3 • SELECT new_subject_id = MAX(subject_id) + 1 FROM subjects;
4
5 • INSERT INTO subjects(subject_id, subject_name)
6     VALUES(new_subject_id, "new_subject_name");
7
8 • INSERT INTO teachers(teacher_id, teacher_name, age, subject_id, school_id)
9     VALUES(id_teacher, "name_teacher", new_subject_id, school_id);
10
11 • INSERT INTO scores(student_id, subject_id, oral_score, midterm_score, final_score)
12     VALUES(student_id, new_subject_id, oral_score, midterm_scores, final_score);
13
14 • COMMIT;
```

3. Xóa một học sinh ra khỏi CSDL:

Các bước thực hiện

- Bắt đầu Transaction
- Kiểm tra học sinh có tồn tại hay không. Nếu không tồn tại → Rollback
- Xóa dữ liệu học sinh với ID học sinh cần xóa trong các bảng attendance, parents, scores, students, tuition_fee
- Thực hiện commit

3. Xóa một học sinh ra khỏi CSDL:

Cài đặt:

```
1 • START TRANSACTION;
2
3 • SELECT @flag := COUNT(student_id) FROM students WHERE student_id = id_xoa_hs;
4
5     IF ( @flag = 0 )
6     THEN ROLLBACK;
7     END IF;
8
9 • DELETE FROM attendance WHERE student_id = id_xoa_hs;
10
11 • DELETE FROM parents WHERE student_id = id_xoa_hs;
12
13 • DELETE FROM scores WHERE student_id = id_xoa_hs;
14
15 • DELETE FROM students WHERE student_id = id_xoa_hs;
16
17 • DELETE FROM tuition_fee WHERE (SELECT students.id FROM students
18                                INNER JOIN tuition_fee WHERE students.id = tuition_fee.id
19                                AND students.student_id = id_xoa_hs);
20
21 • COMMIT;
```

4. Sửa thông tin học sinh trong CSDL:

Các bước thực hiện

- Bắt đầu Transaction
- Kiểm tra học sinh có tồn tại hay không nếu không tồn tại → Rollback
- Sửa thông tin học sinh với ID học sinh cần sửa trong các bảng attendance, parents, scores, students, tuition_fee
- Thực hiện commit

4. Sửa thông tin trong CSDL:

Cài đặt:

```
1 • START TRANSACTION;
2
3 • SELECT @flag := COUNT(student_id) FROM students WHERE student_id = p_student_id;
4
5     IF (@flag = 0)
6     THEN ROLLBACK;
7     END IF;
8
9 • UPDATE attendance SET student_id = p_student_id, num_of_absences = p_num_of_absences
10     WHERE student_id = p_student_id;
11
12 • UPDATE parents SET student_id = p_student_id, phone = p_phone, email = p_email
13     WHERE student_id = p_student_id;
14
15 • UPDATE scores SET student_id = p_studen_id, oral_score = p_oral_score,
16     midterm_score = p_midterm_score, final_score = p_final_score
17     WHERE student_id = p_student_id;
18
19 • UPDATE students SET student_id = p_student_id, student_name = p_student_name,
20     date_of_birth = p_date_of_birth, gender = p_gender, id = p_id
21     WHERE student_id = p_student_id;
22
23 • UPDATE tuition_fee SET id = p_id, fee = p_fee WHERE (SELECT students.id FROM students
24     INNER JOIN tuition_fee WHERE students.id = tuition_fee.id
25     AND students.student_id = id_xoa_hs);
26
27 • COMMIT;
```

5. Cập nhật điểm môn học chi tiết cho một học sinh:

Các bước thực hiện

- Bắt đầu Transaction
- Kiểm tra có tồn tại môn học cụ thể như vậy hay không. Nếu không tồn tại → Rollback
- Cập nhật bảng điểm vào scores
- Đánh giá điểm tổng kết → Cập nhật vào bảng student_rat → Commit

5. Cập nhật điểm môn học chi tiết cho một học sinh:

Cài đặt:

```
1 • START TRANSACTION;
2
3 • SELECT @flag_sb := COUNT(subject_id) FROM scores
4     WHERE p_student_id = student_id AND p_subject_id = subject_id
5         AND p_oral_score = oral_score AND p_midterm_score = midterm_score
6         AND p_final_score = final_score;
7
8 • INSERT INTO scores(student_id, subject_id, oral_score, midterm_score, final_score)
9     VALUES(p_student_id, p_subject_id, p_oral_score, p_midterm_score, p_final_score);
10
11 • SELECT @flag_sc := final_score FROM scores
12     WHERE p_student_id = student_id AND p_subject_id = subject_id
13         AND p_oral_score = oral_score AND p_midterm_score = midterm_score
14         AND p_final_score = final_score;
15
16 IF( @flag_sb = 0 ) THEN
17     ROLLBACK;
18
19 ELSEIF ( @flag_sc != 0 ) THEN
20     INSERT INTO student_rate(student_id, subject_id, oral_score, midterm_score, final_score)
21         VALUES(p_student_id, p_subject_id, p_oral_score, p_midterm_score, p_final_score);
22
23 • COMMIT;
24 ELSE COMMIT;
25 END IF;
```



7.3 SINH CƠ SỞ DỮ LIỆU

7.3.PROCEDURE COPY ADDRESS:

```
SET ANSI_NULLS ON
```

```
GO
```

```
SET QUOTED_IDENTIFIER ON
```

```
GO
```

```
CREATE PROCEDURE copy_address
```

```
AS
```

```
BEGIN
```

```
    INSERT INTO address_copy(address_id, address_name)  
    SELECT address_id, address_name  
    FROM address
```

```
END
```

```
GO
```

SQLQuery4.sql - T...DUCATION (sa (65))* SQLQuery3.sql - T...DUCATION (sa (65))

copy_address
SELECT * FROM address_copy

200 %

Results Messages

	address_id	address_name
1	1	Ba Đình
2	2	Đồng Đa
3	3	Hai Bà Trưng
4	4	Hoàn Kiếm
5	5	Tây Hồ
6	6	Thanh Xuân
7	7	Hoàng Mai
8	8	Long Biên
9	9	Cầu Giấy
10	10	Bắc Từ Liêm
11	11	Nam Từ Liêm
12	12	Hà Đông
13	13	Thanh Trì
14	14	Đống Anh
15	15	Thường Tín
16	16	Mê Linh
17	17	Sóc Sơn
18	18	Thạch Thất
19	19	Quốc Oai
20	20	Thanh Oai
21	21	Đan Phượng
22	22	Gia Lâm

	address_id	address_name
1	1	Ba Đình
2	2	Đồng Đa
3	3	Hai Bà Trưng
4	4	Hoàn Kiếm
5	5	Tây Hồ
6	6	Thanh Xuân
7	7	Hoàng Mai
8	8	Long Biên
9	9	Cầu Giấy
10	10	Bắc Từ Liêm
11	11	Nam Từ Liêm
12	12	Hà Đông
13	13	Thanh Trì
14	14	Đống Anh
15	15	Thường Tín
16	16	Mê Linh
17	17	Sóc Sơn
18	18	Thạch Thất
19	19	Quốc Oai
20	20	Thanh Oai
21	21	Đan Phượng
22	22	Gia Lâm

KẾT LUẬN

Đã xây dựng được một cơ sở dữ liệu thỏa mãn yêu cầu bài toán:

1. Chuẩn hóa CSDL đến chuẩn 3NF.
2. Có 11 câu truy vấn và 7 đại số quan hệ.
3. Thực hiện tối ưu truy vấn bằng tối ưu câu lệnh, INDEX và PARTITION.
4. Tạo được 55 PROCEDURE để select, insert, update, delete, copy từng bảng dữ liệu.
5. Tạo được 5 nghiệp vụ thực tế.
6. Có PROCEDURE đổ dữ liệu từ CSDL cũ sang CSDL mới giống hệt cấu trúc.

**Thanks
for watching**

The background features two solid blue geometric shapes on the right side. A dark blue rectangle is positioned in the top right corner. A lighter blue rectangle is positioned below it, extending from the bottom edge of the frame and partially overlapping the dark blue rectangle.