

UNIVERSIDAD DE SONORA

Semestre: Tecero

Carrera: Ingeniería en Sistemas de Información

Tarea: Consultas a la base de datos de una universidad

Maestro: René Francisco Navarro Hernández

Alumno: Sanchez Peralta Eduardo Sebastian

Consulta 1: Obtener el nombre y el ID de todos los alumnos.

π name, ID (student)

Execution time: 0 ms

student.name	student.ID
'Zhang'	128
'Shankar'	12345
'Brandt'	19991
'Chavez'	23121
'Peltier'	44553
'Levy'	45678
'Williams'	54321
'Sanchez'	55739
'Snow'	70557
'Brown'	76543
'Aoi'	76653
'Bourikas'	98765
'Tanaka'	98988

Consulta 2: Obtener el nombre y DNI de los alumnos que tienen más de 59 créditos.

$\pi_{\text{name, ID}} (\sigma_{\text{tot_cred} > 59} (\text{student}))$

Execution time: 1 ms

student.name	student.ID
'Zhang'	128
'Brandt'	19991
'Chavez'	23121
'Aoi'	76653
'Bourikas'	98765
'Tanaka'	98988

Consulta 3: Obtener el nombre y la identificación de los estudiantes del departamento de ciencias de la computación (Comp. Sci.) que tengan entre 50 y 100 créditos.

Π name, ID (σ dept_name = 'Comp. Sci.' and tot_cred \geq 50 and tot_cred \leq 100 (student))

Execution time: 1 ms

student.name	student.ID
'Williams'	54321
'Brown'	76543

Consulta 4: Obtener todos los datos de los cursos ofrecidos en el departamento de biología.

$\sigma_{\text{dept_name} = \text{'Biology'}} (\text{course})$

Execution time: 2 ms

course.course_id	course.title	course.dept_name	course.credits
'BIO-101'	'Intro. to Biology'	'Biology'	4
'BIO-301'	'Genetics'	'Biology'	4
'BIO-399'	'Computational Biology'	'Biology'	3

Consulta 5: Obtenga el nombre y el ID de los cursos que tienen 4 créditos.

Π title, course_id (σ credits = 4 (course))

Execution time: 2 ms

course.title	course.course_id
'Intro. to Biology'	'BIO-101'
'Genetics'	'BIO-301'
'Intro. to Computer Science'	'CS-101'
'Game Design'	'CS-190'
'Physical Principles'	'PHY-101'

Consulta 6: Obtener nombre e identificación de los cursos del departamento de biología o historia.

Π title, course_id (σ dept_name = 'Biology' or dept_name = 'History' (course))

Execution time: 2 ms

course.title	course.course_id
'Intro. to Biology'	'BIO-101'
'Genetics'	'BIO-301'
'Computational Biology'	'BIO-399'
'World History'	'HIS-351'

Consulta 7: Obtener los nombres de los departamentos que ofrecen cursos de 4 o 3 créditos.

$\pi_{dept_name} (\sigma_{credits = 4 \text{ or } credits = 3} (course))$

Execution time: 1 ms

course.dept_name

'Biology'

'Comp. Sci.'

'Elec. Eng.'

'Finance'

'History'

'Music'

'Physics'

Consulta 8: Obtener los nombres de los departamentos que ofrecen cursos de 4 y 3 créditos.

$$\pi_{\text{dept_name}} (\sigma_{\text{credits} = 4} (\text{course})) \cap \pi_{\text{dept_name}} (\sigma_{\text{credits} = 3} (\text{course}))$$

Execution time: 2 ms

course.dept_name

'Biology'

'Comp. Sci.'

Consulta 9: Obtenga una lista de los cursos impartidos por cada instructor (nombre del instructor, ID del curso).

Π instructor.name, teaches.course_id (instructor \bowtie instructor.ID = teaches.ID (teaches))

Execution time: 1 ms

instructor.name	teaches.course_id
'Srinivasan'	'CS-101'
'Srinivasan'	'CS-315'
'Srinivasan'	'CS-347'
'Wu'	'FIN-201'
'Mozart'	'MU-199'
'Einstein'	'PHY-101'
'El Said'	'HIS-351'
'Katz'	'CS-101'
'Katz'	'CS-319'
'Crick'	'BIO-101'
'Crick'	'BIO-301'
'Brandt'	'CS-190'
'Brandt'	'CS-319'
'Kim'	'EE-181'

Consulta 10: Obtenga una lista de los cursos, del semestre de otoño, impartidos por cada instructor (nombre del instructor, ID del curso).

```
 $\Pi$  instructor.name, teaches.course_id (  $\sigma$  semester = 'Fall' ( instructor  $\bowtie$  instructor.ID = teaches.ID ( teaches ) ) )
```

Execution time: 1 ms

instructor.name	teaches.course_id
'Srinivasan'	'CS-101'
'Srinivasan'	'CS-347'
'Einstein'	'PHY-101'

Consulta 11: Obtenga una lista de cursos del semestre de primavera impartidos por instructores con un salario de menos de \$ 80,000 (nombre del instructor, ID del curso).

```

$$\Pi_{\text{instructor.name, teaches.course\_id}} \left( \sigma_{\text{semester} = \text{'Spring'} \text{ and } \text{salary} < 80000} \left( \text{instructor} \bowtie \text{instructor.ID} = \text{teaches.ID} \left( \text{teaches} \right) \right) \right)$$

```

Execution time: 1 ms

instructor.name	teaches.course_id
'Srinivasan'	'CS-315'
'Mozart'	'MU-199'
'El Said'	'HIS-351'
'Katz'	'CS-101'
'Katz'	'CS-319'

Consulta 13: Obtenga el nombre del estudiante y el nombre del curso de los estudiantes del departamento de ciencias de la computación que tomaron cursos en 2009.

```
 $\Pi$  student.name, course.title (  $\sigma$  student.dept_name = 'Comp. Sci.' and takes.year = 2009 ( ( student  $\bowtie$  student.ID = takes.ID ( takes ) )  $\bowtie$  takes.course_id = course.course_id ( course ) ) )
```

Execution time: 1 ms

student.name	course.title
'Zhang'	'Intro. to Computer Science'
'Zhang'	'Database System Concepts'
'Shankar'	'Intro. to Computer Science'
'Shankar'	'Game Design'
'Shankar'	'Database System Concepts'
'Williams'	'Intro. to Computer Science'
'Williams'	'Game Design'
'Brown'	'Intro. to Computer Science'

Consulta 14: Obtener los nombres de los cursos y las calificaciones de los cursos aprobados por el estudiante 'Chávez'.

```
 $\Pi$  course.title, takes.grade (  $\sigma$  student.name = 'Chávez' and takes.grade  $\neq$  'F' (
  ( student  $\bowtie$  student.ID = takes.ID ( takes ) )  $\bowtie$  takes.course_id =
    course.course_id ( course ) ) )
```

Execution time: 2 ms

course.title	takes.grade
--------------	-------------

Consulta 15: Obtenga la lista de cursos que no están serializados con otro curso.

$\pi_{\text{course_id}, \text{title}} (\text{course}) - \pi_{\text{course.course_id}, \text{course.title}} (\text{course} \bowtie \text{course.course_id} = \text{prereq.course_id} (\text{prereq}))$

Execution time: 2 ms

course.course_id	course.title
'BIO-101'	'Intro. to Biology'
'CS-101'	'Intro. to Computer Science'
'FIN-201'	'Investment Banking'
'HIS-351'	'World History'
'MU-199'	'Music Video Production'
'PHY-101'	'Physical Principles'

Consulta 16: Obtenga una lista de los nombres de los estudiantes que tienen un asesor.

$\pi_{\text{name}} (\text{student} \bowtie_{\text{advisor.s_id} = \text{advisor.s_id}} (\text{advisor}))$

Execution time: 2 ms

student.name
'Zhang'
'Shankar'
'Brandt'
'Chavez'
'Peltier'
'Levy'
'Williams'
'Sanchez'
'Snow'
'Brown'
'Aoi'
'Bourikas'
'Tanaka'

Consulta 17: Obtenga una lista de los nombres de los estudiantes que no tienen un asesor.

```
 $\pi_{\text{name}}(\text{student}) - \pi_{\text{student.name}}(\text{student} \bowtie_{\text{advisor.s\_id} = \text{advisor.s\_id}}(\text{advisor}))$ 
```

Execution time: 1 ms

student.name

Consulta 18: Obtenga una lista con el nombre del estudiante y el nombre del asesor para estudiantes de biología.

```
 $\pi$  student.name, instructor.name (  $\sigma$  student.dept_name = 'Biology' ( ( student  $\bowtie$  advisor.s_id = advisor.s_id ( advisor ) )  $\bowtie$  advisor.i_id = instructor.ID ( instructor ) ) )
```

Execution time: 2 ms

student.name	instructor.name
'Tanaka'	'Katz'
'Tanaka'	'Srinivasan'
'Tanaka'	'Singh'
'Tanaka'	'Einstein'
'Tanaka'	'Kim'
'Tanaka'	'Crick'

Consulta 19: Mostrar las aulas asignadas al departamento de Biología.

```
 $\Pi$  building, department.dept_name (  $\sigma$  dept_name = 'Biology' ( department ) )
```

Execution time: 2 ms

department.building	department.dept_name
'Watson'	'Biology'

Consulta 20: Obtener el número de cursos que ofrece cada departamento.

```
Y dept_name; count(course_id) →total_cursos(course department)
```

course.dept_name	total_cursos
'Biology'	3
'Comp. Sci.'	5
'Elec. Eng.'	1
'Finance'	1
'History'	1

course.dept_name	total_cursos
'Music'	1
'Physics'	1

Consulta 21: Obtener el número promedio de cursos ofrecidos por cada departamento por año.

```
1 ȳdept_name; AVG(tot_cursos) → promedio_curso (ȳdept_name, year;  
COUNT(course_id) → tot_cursos(courseȳdepartmentȳsection))
```

course.dept_name	promedio_curso
'Comp. Sci.'	2
'Elec. Eng.'	1
'History'	1
'Music'	1
'Physics'	1

Consulta 22: Consigue el departamento con mayor presupuesto.

```
1 ȳdept_name; AVG(tot_cursos) → promedio_curso (ȳdept_name, year;  
COUNT(course_id) → tot_cursos(courseȳdepartmentȳsection))
```

department.dept_name	department.budget
'Finance'	120000

Consulta 23: Obtener el curso que ha cursado el mayor número de alumnos.

```
1 π course_id, num_alumnos( σ num_alumnos = max_num ( (course ⋈ (γ course_id  
; COUNT(ID)→num_alumnos (takes))) ⋈ (γ ; MAX(num_alumnos)→max_num (γ  
course_id ; COUNT(ID)→num_alumnos (takes))))
```

course.course_id	num_alumnos
'CS-101'	7

Consulta 24: Obtener el salario promedio de los instructores.

```
Y ; avg(salary) → salario_promedio (instructor)
```

salario_promedio
74833.33333333333

Consulta 25: Consiga instructores que tengan un salario más alto que el salario promedio.

```
1 π name, salary (σ (salary > avg_sal) (instructor ⋈ ( γ ; AVG(salary) → avg_sal (instructor))))
```

instructor.name	instructor.salary
'Wu'	90000
'Einstein'	95000
'Gold'	87000
'Katz'	75000
'Singh'	80000
'Brandt'	92000
'Kim'	80000