# Laboratorio de Computación II

**Examen 2do Parcial**

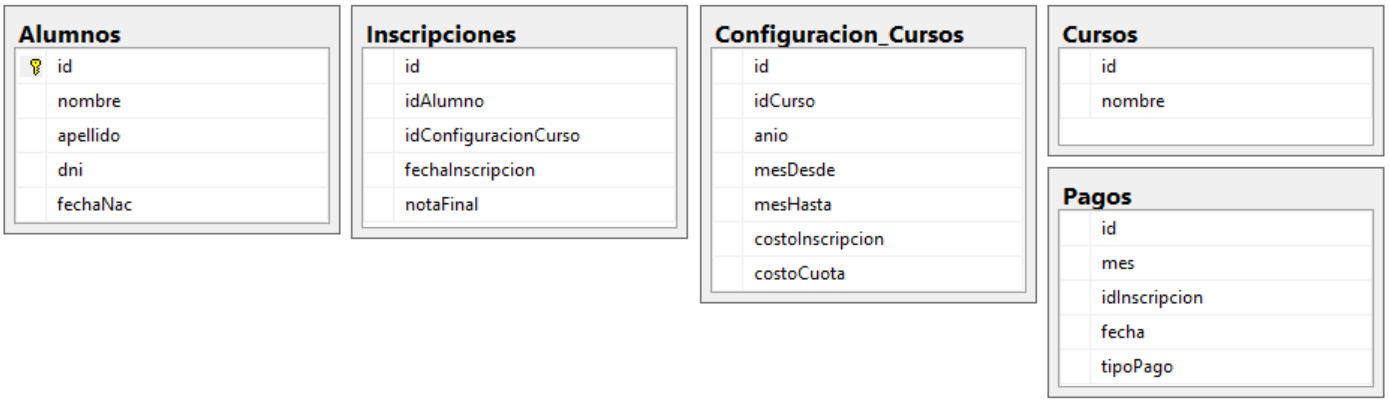
**Comisión 1W3**

**29/10/2019**

| Legajo | Apellido y Nombre |
|---|---|
| | |

**[ 90%] SQL**

**Tablas:**




**Resolver las siguientes consultas:**

| | | |
|---|---|---|
| | 20% | Mostrar nombre, apellido y nombre del curso de los alumnos que esten cursando en el año actual y que no tengan ninguna cuota paga. |
| | 20% | Mostrar los cursos, la cantidad de cuotas pagas y el monto total recaudado en cuotas (considerando que la columna tipoPago de la tabla pagos asume el valor 1 para costo de inscripción y valor 2 para costo de cuota) de los mismos por año pero solo de aquellos donde el promedio recaudado en cuotas para ese año sea mayor al promedio general de los cursos para el mismo año. |
| | 15% | Crear una vista que devuelva los siguientes campos: id de curso, nombre del curso y promedio de notas del curso por año. Descartar los cursos que tienen menos de 15 alumnos por año. |
| | 20% | Crear un procedimiento almacenado que reciba un año por parámetro y muestre nombre del curso y promedio de notas del curso que haya tenido el mayor promedio de notas en ese año pasado por parámetro. Utilizar la vista creada en el punto anterior para resolver este ejercicio. |
| | 15% | Diseñe, basado en la base de datos del parcial, una función indicando que tipo de función es. Debe tener la consigna o enunciado y la función debe responder a dicha consigna |

**[ 10%] TEÓRICO**

A. Explique qué es una referencia externa en una subconsulta
B. Explique el test de existencia EXISTS. Como es la sintaxis básica del test y qué requisitos debe cumplir la subconsulta con la que se utiliza dicho test?

| Tema | SQL | Teórico | Total | Nota |
|---|---|---|---|---|
| *1* | | | | |

1)

```
SELECT a.nombre, a.apellido, c.nombre 'Curso'
FROM Alumnos a
JOIN Inscripciones i ON a.id = i.idAlumno
JOIN Configuracion_Cursos cc ON i.idConfiguracionCurso = cc.id
JOIN Cursos c ON cc.idCurso = c.id
WHERE cc.anio = YEAR(GETDATE())
AND NOT EXISTS (SELECT * FROM Pagos p
                WHERE p.idInscripcion = i.id
                AND p.tipoPago = 2)
```

2)

```
SELECT c.nombre 'Curso', COUNT(*) 'Cantidad', SUM(cc.costoCuota) 'MontoTotal', cc.anio
FROM Inscripciones i
JOIN Pagos p ON i.id = p.idInscripcion
JOIN Configuracion_Cursos cc ON i.idConfiguracionCurso = cc.id
JOIN Cursos c ON cc.idCurso = c.id
WHERE p.tipoPago = 2
GROUP BY c.nombre, cc.anio
HAVING AVG(cc.costoCuota) > (SELECT AVG(cc1.costoCuota)
                             FROM Configuracion_Cursos cc1
                             JOIN Inscripciones i1 ON cc1.id = i1.idConfiguracionCurso
                             JOIN Pagos p1 ON i1.id = p1.idInscripcion
                             WHERE cc1.anio = cc.anio
                             AND p1.tipoPago = 2)
```

3)

```
CREATE VIEW NombreVista
AS
SELECT c.id 'idCurso', c.nombre 'curso', AVG(i.notaFinal) 'promedio', cc.anio
FROM Alumnos a
JOIN Inscripciones i ON a.id = i.idAlumno
JOIN Configuracion_Cursos cc ON i.idConfiguracionCurso = cc.id
JOIN Cursos c ON cc.idCurso = c.id
GROUP BY c.id, c.nombre, cc.anio
HAVING COUNT(*) >= 15
```

4)

```
CREATE PROCEDURE NombreProcedimiento
@anio int
AS
SELECT v.curso, v.promedio
FROM NombreVista v
WHERE v.anio = @anio
AND v.promedio = (SELECT MAX(v1.promedio)   FROM   NombreVista v1  WHERE v1.anio = @anio)
```

# Laboratorio de Computación II

**Examen 2do Parcial**

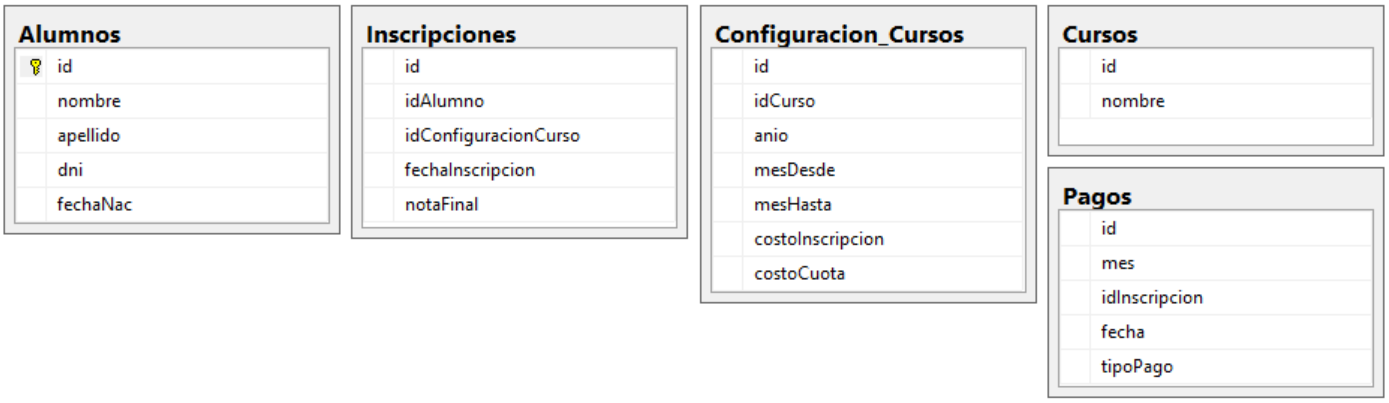
**Comisión 1W3**

**29/10/2019**

| Legajo | Apellido y Nombre |
|---|---|
| | |

**[ 90%] SQL**

**Tablas:**




**Resolver las siguientes consultas:**

| | | |
|---|---|---|
| | 20% | Mostrar nombre, apellido y nombre del curso de los alumnos que hayan aprobado todos los cursos que hayan realizado con notas mayores a 7 |
| | 20% | Mostrar los cursos, la cantidad de alumnos inscriptos y el promedio de notas de dicho curso por año pero solo de aquellos donde el promedio de notas para ese año sea menor o igual al promedio general de notas de los cursos para el mismo año. |
| | 20% | Crear una vista que devuelva los siguientes campos: id de inscripcion, datos del alumno, nombre de curso y mes de cuota pagada pero solo de las cuotas que hayan sido pagadas fuera de término (o sea que la fecha de pago tenga un mes mayor al mes de la cuota que se pago) y de los alumnos que ya tengan paga la inscripcion (considerando que la columna tipoPago de la tabla pagos asume el valor 1 para costo de inscripción y valor 2 para costo de cuota). |
| | 15% | Crear un procedimiento almacenado que reciba un id de inscripción por parámetro y muestre los datos del alumno, y el monto total pagado (incluir cuotas y monto de inscripción en el monto total) del alumno que pertenece al id de inscripción pasado por parámetro. Utilizar la vista creada en el punto anterior para resolver este ejercicio. |
| | 15% | Diseñe, basado en la base de datos del parcial, una función indicando que tipo de función es. Debe tener la consigna o enunciado y la función debe responder a dicha consigna |

**[ 10%] TEÓRICO**

A. Explique qué es una referencia externa en una subconsulta
B. Explique el test de existencia EXISTS. Como es la sintaxis básica del test y qué requisitos debe cumplir la subconsulta con la que se utiliza dicho test?

| Tema | SQL | Teórico | Total | Nota |
|---|---|---|---|---|
| *2* | | | | |

1)

```
SELECT a.nombre, a.apellido, c.nombre 'Curso'
FROM Alumnos a
JOIN Inscripciones i ON a.id = i.idAlumno
JOIN Configuracion_Cursos cc ON i.idConfiguracionCurso = cc.id
JOIN Cursos c ON cc.idCurso = c.id
WHERE 7 < ALL (SELECT i1.notaFinal FROM Inscripciones i1
                WHERE i1.idAlumno = a.id)
```

2)

```
SELECT c.nombre 'Curso', COUNT(*) 'Cantidad', AVG(i.notaFinal) 'Promedio', cc.anio
FROM Inscripciones i
JOIN Configuracion_Cursos cc ON i.idConfiguracionCurso = cc.id
JOIN Cursos c ON cc.idCurso = c.id
GROUP BY c.nombre, cc.anio
HAVING AVG(i.notaFinal) <= (SELECT AVG(i1.notaFinal)
                            JOIN Inscripciones i1
                            WHERE YEAR(i1.fechaInscripcion) = cc.anio)
```

3)

```
CREATE VIEW NombreVista
AS
SELECT i.id 'idInscripcion', a.nombre, a.apellido, c.nombre 'curso', p.mes
FROM Alumnos a
JOIN Inscripciones i ON a.id = i.idAlumno
JOIN Configuracion_Cursos cc ON i.idConfiguracionCurso = cc.id
JOIN Cursos c ON cc.idCurso = c.id
JOIN Pagos p ON i.id = i.idInscripcion
WHERE p.mes < MONTH(p.fecha)
AND p.tipoPago = 2
AND EXISTS (SELECT * FROM Pagos p1 WHERE p1.tipoPago = 1 AND p1.idInscripcion = i.id)
```

4)

```
CREATE PROCEDURE NombreProcedimiento
@idInscripcion int
AS
BEGIN
	DECLARE @montoCuotas decimal(8,2)
	DECLARE @montoInscripcion decimal(8,2)

	SELECT @montoInscripcion = SUM(cc.costoInscripcion)
	FROM Pagos p
	JOIN Inscripciones i ON p.idInscripcion = i.id
	JOIN Configuracion_Cursos cc ON i.idConfiguracionCurso = cc.id
	WHERE p.idInscripcion = @idInscripcion
	AND p.tipoPago = 1

	SELECT @montoCuotas = SUM(cc.costoCuota)
	FROM Pagos p
	JOIN Inscripciones i ON p.idInscripcion = i.id
	JOIN Configuracion_Cursos cc ON i.idConfiguracionCurso = cc.id
	WHERE p.idInscripcion = @idInscripcion
	AND p.tipoPago = 2

	SELECT v.nombre, v.apellido, @montoInscripcion + @montoCuotas 'Monto Total'
	FROM NombreVista v
	WHERE v.idInscripcion = @idInscripcion
END
```