|  |  |  |  |
| --- | --- | --- | --- |
| **Profesor:** | FRANCISCO JAVIER CALLE GOMEZ | **Grupo** | 81 |
| **Alumno/a:** | JORGE RODRÍGUEZ FRAILE | **NIA:** | 100405951 |
| **Alumno/a:** | CARLOS RUBIO OLIVARES | **NIA:** | 100405834 |

# Introducción

Una introducción que sea el punto de partida del trabajo y sirva para analizar el problema que se va a resolver, fije los objetivos que se persiguen, y describa la estructura del documento.

# Consultas

**Burden-user:**

1. su diseño en álgebra relacional
2. **SELECT**
3. a.nick,
4. a.reg\_date,
5. b.citizenid
6. **FROM**
7. users a
8. **LEFT** **JOIN** profiles b **ON** (a.nick = b.nick)
9. **WHERE** (months\_between (sysdate, a.reg\_date) >= 6
10. **AND** a.nick **NOT** **IN** (
11. **SELECT**
12. nick
13. **FROM**
14. membership)
15. **AND** (b.citizenID **IS** **NULL**
16. **OR** b.citizenID **NOT** **IN** (
17. **SELECT**
18. citizenID
19. **FROM**
20. contracts)));
21. las pruebas realizadas para demostrar que funciona correctamente

**FilmMaster:**

1. su diseño en álgebra relacional
2. **SELECT**
3. dic,
4. avg(c)
5. **FROM** (
6. **SELECT**
7. a.director dic,
8. a.title,
9. **COUNT**('x') c
10. **FROM**
11. movies a
12. **JOIN** comments b **ON** (a.director = b.director)
13. **GROUP** **BY**
14. a.director,
15. a.title)
16. **GROUP** **BY**
17. dic
18. **ORDER** **BY**
19. avg(c)
20. **DESC** fetch **FIRST** **ROW** **ONLY**;
21. Observando el mas coments y su num pelis, hacer la medio y ver si funka.

# Vistas

Incluye una subsección por cada vista que desarrolles (al menos una, si bien puedes resolver más como trabajo opcional). Para cada vista, debes exponer:

1. su diseño en álgebra relacional
2. su implementación en SQL
3. Pruebas: no sólo se debe comprobar que la vista está bien definida (como una consulta), sino también la funcionalidad de la vista: consulta (índica el número de filas del resultado), inserción, borrado y modificación (es necesario establecer qué operaciones resuelve el gestor y qué operaciones no; pero no es necesario implementar las que el gestor no resuelva).

# Tablas Históricas

Describir qué elementos se ha estimado oportuno para recoger esta semántica excluida (vistas, disparadores, tablas …). Se detallará el diseño de cada uno de los elementos incorporados, su implementación (código SQL), y pruebas. En caso de utilizar vistas, estas deberán ser completamente funcionales (según se describe en el enunciado), y los elementos introducidos para completar esa funcionalidad también deberán ser descritos, implementados y probados.

# Diseño Externo

Describe cada privilegio que debe asignarse al rol correspondiente, y cuál/es requiere/n de la creación de una o más vistas. Describe esas vistas y realiza su diseño, implementación y pruebas (de modo análogo a como se hiciera en el apartado 3).

# Disparadores

Para cada disparador resuelto se debe incluir una subsección que contenga:

* Descripción del diseño: Tabla a la que está asociado, Evento o eventos en los que se dispara, Temporalidad (antes, después o en vez de), Granularidad (por fila o sentencia), Condición (si la tiene) y Acción (descripción en lenguaje natural).
* Código (en SQL)
* Pruebas

# Conclusiones

En primer lugar, debéis defender el resultado que hayáis alcanzado, haciendo hincapié en la cobertura semántica y potencia de vuestra implementación.

Después, comentad vuestro desempeño en estas dos primeras prácticas: esfuerzo requerido, organización de vuestro equipo de trabajo, progreso en vuestros conocimientos, etc. También podéis proponer mejoras para otros años (tamaño del problema, elementos que se piden, valoración, plazos, material de apoyo, etc).