

DIPLOMATURA EN SOFTWARE LIBRE

CURSO: Bases de Datos PostgreSQL

Examen

Institución: Universidad del Este - La Plata - Buenos Aires – Argentina

Período: 1º CUATRIMESTRE de 2015

Docente coordinador: Mg. Lic. Mariano Reingart

Observaciones Generales

- La entrega se realizará por el campus virtual de la universidad, preferentemente antes del 19 de Julio de 2015.
- El trabajo debe ser original, individual y personal, ante cualquier tipo de plagio se invalida el examen y se comunicará la situación a las autoridades de la universidad para que tomen las medidas correspondientes.
- No se aceptarán escritos sin referencias a fuentes relevantes, confiables y verificables. En especial, utilizar Wikipedia o sitios similares sólo como consulta preliminar¹. Limite de longitud: ver ODT (tamaño máximo: 1MB).
- Para aprobar el examen se debe alcanzar una calificación de 70 puntos. La entrega fuera de fecha quedará a criterio de los docentes, y debe estar debidamente justificada (tanto en las cuestiones personales como en la calidad de los contenidos), o se descontarán puntos por retraso.

Guía de Examen

Estudio de caso (100 puntos)

Seleccione una proyecto de software libre, un conjunto de datos públicos / abiertos o una base de datos tentativa, por ejemplo:

- <http://data.buenosaires.gob.ar/dataset/elecciones-2013>
- <http://www.ec.gba.gov.ar/estadistica/censo2010/cartografia.html>
<http://www.ec.gba.gov.ar/estadistica/>
- <http://www.indec.gov.ar/bases-de-datos.asp>
- <http://trac.usla.org.ar/sanca> (SancaBase) o cualquier otro proyecto libre

IMPORTANTE: Presentar la base de datos elegida en el Foro General del curso para evitar solapamiento y confirmar que sea viable antes de iniciar el trabajo.

¹ <http://isites.harvard.edu/icb/icb.do?keyword=k70847&pageid=icb.page346376>

Realizar un documento similar al “[Caso de Estudio](#)²” (licencia GNU FDL 1.2+ para su eventual publicación) contemplando (10 puntos c/u):

1. Presentación y relevamiento del caso de estudio
2. Diagrama Entidad-Relación y transformación al Modelo Relacional (normalizado)
3. Esquema Relacional y Álgebra relacionar
4. Lenguaje SQL
 5. Sentencias DDL de creación de tablas
 6. Sentencias DML de manipulación de datos.
7. Análisis comparativo PostgreSQL vs MySQL, FireBird, Oracle, MS SQL Server, u otro (licenciamiento, soporte, comunidad, características, valoración, etc.)
8. Consideraciones implementación en PostgreSQL (respaldo y restauración, tipos de datos, seguridad, integridad, procedimientos almacenados, índices y análisis de rendimiento, mantenimiento, etc.)
9. Desarrollo de aplicación en Python con conexión a la base de datos (importación, procesamiento, búsqueda, reportes, visualización, exportación, o similar)
10. Análisis DAFO y conclusiones finales

Adicionalmente al apunte de UOC, se pide (de ser posible):

- Los datos iniciales deberían poder importarse con COPY, y luego utilizar INSERT / UPDATE / DELETE para normalizar esos datos en sus respectivas tablas. El esquema inicial debe contener todos los campos, y luego deben ejecutarse los ALTER / DROP TABLE para eliminar los campos o tablas redundantes.
- Realizar y crear una vista para cada consultas SQL relevantes
- Resultado de la ejecución de dichos ejercicios (incluir SQL y datos devueltos -ej-)
- Convertir las consultas de SQL a Álgebra Relacional
- Definir las Dependencias Funcionales y Normalizar intuitivamente (3FN o FNBC)
- Análisis de ejecución de consulta (EXPLAIN ANALYZE gráfico explicando cada nodo) de una consulta no trivial. Indicar que índices se podrían agregar (SQL).
- Integridad, Validaciones y Restricciones: claves primarias (PRIMARY KEY) y foráneas (FOREIGN KEY), no nulo (NOT NULL), (CHECK), política de actualización: en cascada (CASCADE), política de eliminación: asignación de valores nulos (SET NULL), que correspondan
- Disparador (trigger) para actualizar automáticamente campos calculados o similares
- SQL de Creación de Usuarios (por ej. CREATE USER) y SQL para otorgar/denegar Permisos de Acceso (GRANT/REVOKE):
- En el caso de los campos calculados, recomendaciones a realizar al desarrollador cuando tenga que consultarlos o actualizarlos. Indicar casos de uso para SELECT ... FOR UPDATE/SHARE y LOCK TABLE.
- Realizar un pequeño script SQL de mantenimiento, que limpie las filas obsoletas y actualice las estadísticas (usando VACUUM). Ejecútelo y comente la salida.

2 <http://ocw.uoc.edu/informatica-tecnologia-y-multimedia/bases-de-datos/materiales/>