**Considere las siguientes especificaciones para la entrega de evidencias del curso de bases de Datos Distribuidas, las sugerencias las preguntas tipo en el examen presencial, tomadas del libro Mastering Oracle SQL, 2nd Edition, para lo cual habrá de instalar la base de datos correspondiente, tomando en cuenta los scripts, que lo acompañan.**

|  |  |  |
| --- | --- | --- |
| **1** | **Entrega de las evidencias de los ejercicios de consultas y del video elaborado.** | **Rúbricas** |
| **2** | **Entrega del primer examen escrito referente a conceptos de bases distribuidas** |  |
| **3** | **Entrega en carpetas de archivos, por alumno, por capítulo, separados por tipo o dominio de archivo, txt o sql. En archivos de scripts, de lanzadores, y archivos spool.** |  |
| **4** | **Entrega los ejercicios de SQL by Example, PL/SQL programming , Triggeres and store procedures CAP 6 Y 7, Cost Based Oracle y VIDEO de acuerdo a las rúbricas establecidas en clase.** |  |
| **5** | **Presentarse al examen presencial individual.** |  |
| **6** | **Revisar los ejercicios de consultas del libro de Mastering Oracle ed Oreilly en sus capítulos 8,9,14..** |  |
| **7** | **Entrega del examen escrito referente a temas del libro Oracle® PL/SQL™ by Example** |  |
| **8** | **Entrega de los ejercicios del libro Cost-Based Oracle Fundamentals** |  |
| **9** | **Elaborar un reporte de los video de Jeniffer Widom, profesora de la Universidad de Stanford, con el tema “multivalued-dependencies and 4th Normal Form”,**  **https://www.youtube.com/watch?v=698GsCsmXIk&list=PLroEs25KGvwzmvIxYHRhoGTz9w8LeXek0&index=23**  **Así como el tema “Boyce\_Codd Normal Form”**  **https://www.youtube.com/watch?v=mfVCesoMaGA&list=PLroEs25KGvwzmvIxYHRhoGTz9w8LeXek0&index=22.**  **El tema de Indices**  **https://www.youtube.com/watch?v=WrKAlmGiipE&list=PLroEs25KGvwzmvIxYHRhoGTz9w8LeXek0&index=32** | **Máximo una cuartilla.** |
| **10** | **Consultas tipo examen:**  **SELECT cust\_nbr, name**  **FROM customer**  **WHERE region\_id = 5**  **UNION**  **SELECT c.cust\_nbr, c.name**  **FROM customer c**  **WHERE c.cust\_nbr IN (SELECT o.cust\_nbr**  **FROM cust\_order o, employee e**  **WHERE o.sales\_emp\_id = e.emp\_id**  **AND e.lname = 'MARTIN');** | **Mastering SQL**  **Chapter 7. Set Operations** 7.1 Set Operators |
| **11** | **SELECT cust\_nbr, name**  **FROM customer**  **WHERE region\_id = 5**  **INTERSECT**  **SELECT c.cust\_nbr, c.name**  **FROM customer c**  **WHERE c.cust\_nbr IN (SELECT o.cust\_nbr**  **FROM cust\_order o, employee e**  **WHERE o.sales\_emp\_id = e.emp\_id**  **AND e.lname = 'MARTIN');** | **Mastering SQL** |
| **12** | **SELECT e.lname "Employee", m.lname "Manager"**  **FROM employee e, employee m**  **WHERE e.manager\_emp\_id = m.emp\_id;**  **SELECT e.lname "Employee", m.lname "Manager"**  **FROM employee e LEFT OUTER JOIN employee m**  **ON e.manager\_emp\_id = m.emp\_id ;**  **SELECT emp\_id, lname, dept\_id, manager\_emp\_id, salary, hire\_date**  **FROM employee e**  **WHERE emp\_id NOT IN**  **(SELECT manager\_emp\_id FROM employee**  **WHERE manager\_emp\_id IS NOT NULL);**  **SELECT e\_top.lname, e\_2.lname, e\_3.lname, e\_4.lname**  **FROM employee e\_top LEFT OUTER JOIN employee e\_2**  **ON e\_top.emp\_id = e\_2.manager\_emp\_id**  **LEFT OUTER JOIN employee e\_3**  **ON e\_2.emp\_id = e\_3.manager\_emp\_id**  **LEFT OUTER JOIN employee e\_4**  **ON e\_3.emp\_id = e\_4.manager\_emp\_id**  **WHERE e\_top.manager\_emp\_id IS NULL;**  **SELECT lname, emp\_id, manager\_emp\_id**  **FROM employee**  **START WITH manager\_emp\_id IS NULL**  **CONNECT BY PRIOR emp\_id = manager\_emp\_id;**  **SELECT lname, emp\_id, manager\_emp\_id**  **FROM employee**  **START WITH lname = 'JONES'**  **CONNECT BY manager\_emp\_id = PRIOR emp\_id;** | **Mastering SQL**  **Chapter 8. Hierarchical Queries** 8.1 Representing Hierarchical Information |
| **13** | **SELECT cust\_sales.cust\_nbr cust\_nbr, cust\_sales.region\_id region\_id,**  **cust\_sales.tot\_sales cust\_sales, region\_sales.tot\_sales region\_sales**  **FROM**  **(SELECT o.region\_id region\_id, SUM(o.tot\_sales) tot\_sales**  **FROM orders o**  **WHERE o.year = 2001**  **GROUP BY o.region\_id) region\_sales INNER JOIN**  **(SELECT o.cust\_nbr cust\_nbr, o.region\_id region\_id,**  **SUM(o.tot\_sales) tot\_sales**  **FROM orders o**  **WHERE o.year = 2001**  **GROUP BY o.cust\_nbr, o.region\_id) cust\_sales**  **ON cust\_sales.region\_id = region\_sales.region\_id**  **WHERE cust\_sales.tot\_sales > (region\_sales.tot\_sales \* .2);** | **Mastering SQL**  **14.1 Analytic SQL Overview.** |