**DB\_Admin, Recovery, and Security (4/542) Fall** **2021 Individual Assignments**

Assignment #5 (ASG5) Tuesday 11/2/21

Your task is to use PubCo and DGDB database to develop a discretionary Access strategy.

This is a team assignment and should be done in two parts.

Part 1: To develop a discretionary access strategy for PubCo.

Part 2: To record the plan for discretionary access strategy for PubCo in DGDB.

Part1: Setting up discretionary to be enforced for PubCo (previously referred to as POPCO) database

What do we need:

* Script to create PubCo in Oracle and MSSQL
  + Oracle: **pubco-script-DBcreate**
  + MSSQL**:**
* Employee\_Role\_Dept\_Table worksheet” Role Matrix-PROP-Version2-102121”. This spreadsheet contains the following worksheets

1. Employee Master List: contains information about userID, employee start\_date, employee end\_date, status ID, status name, role for each employee associated with a department, position and department
   * We need to have a script to create users for PubCo
   * Oracle sample to create user:   
     --create a user in Oracle 10g with DBA power then create multiple users   
     drop user pubco CASCADE;   
     create user pubco Identified by pubco default tablespace users temporary tablespace temp;  
     grant connect to pubco ;   
     grant unlimited tablespace to pubco ;  
     grant create table to pubco;  
     grant create role to pubco;  
     grant create procedure to pubco;  
     grant create trigger to pubco;  
     grant create view to pubco;
   * MSSQL sample to create user: TBD by the team
2. Dept-Role-tables worksheet: identify type of access privileges each role of a given department has on PubCO tables
   * Access privileges: A (SELECT) and X (SELECT, UPDATE, INSERT, and DELETE)
3. Position-Dept worksheet: List all position associated with each department
4. Role worksheet: All available roles
   * Oracle Sample to create role: create role Finance - Accountant

What we need to develop

1. Develop a notepad script containing create all USER.
2. Develop a notepad script containing create all ROLE.
3. Develop a notepad script containing every ROLE associated with each table.
4. Develop a notepad script containing every ROLE associated with each user.
5. See sample script for a similar problem (requirement and data is different)
   * File name: “Sample-PubCo-Discretionary-solution-Part1”
6. Develop a notepad script containing role changes that has happened to any user due to status change.
   * Worksheet Emp\_Role\_Changes: identifies start and end date of any role changes due to status change or changing department
7. Run the script to create the discretionary security in MSSQL and validate it thru MSSQL system catalog
8. Run the script to create the discretionary security in Oracle and validate it thru Oracle system catalog

Part 2: Setting up discretionary access strategy plan and maintenance to be enforced for DGDB database

1. Employee Master List: contains information about
   * Database ID and Name
   * Employee last name, first name, userID
   * Employee start\_date, end\_date, status ID and status name
   * Expand the User\_Access\_previlage spareadsheet to include role level access to each employee of every department
2. DGDB should include PubCo database (Database/schema # 5)
   * DGDB\_v4b
3. DGDB should have list of all PubCo tables
4. DGDB tables that will be or could be affected are (check with data model as well)
   * DG\_Roles
   * DG\_Tables
   * DG\_User\_Database
   * DG\_Unit\_Role\_Table
   * DG\_User\_Unit\_Role
5. Create INSERT statements to include PubCO Discretionary Security
6. Append the INSERT statements at the end of DGDB4
7. Run DGDB and make sure there is no error
8. Develop report to show about the addition of PubCo discretionary plan
9. Compare DGDB discretionary plan for PubCo with its implementation in Oracle and MSSQL

Submit the following documents using the D2L's DROP BOX Asg5 on due date

Part1:

A notepad script to create discretionary security (DS) in MSSQL called PubCo\_DS\_MSSQL\_Team#

A notepad script to create discretionary security (DS) in Oracle called PubCo\_DS\_Oracle\_Team#

A word document for the analysis of your development and verification PubCo\_DS\_Report\_Team#

Part2:

A notepad script to create discretionary security in DGDB for Pubco called DGDB\_PubCo\_DS\_Script\_Team#

A word document for the analysis of your development and verification DGDB\_PubCo\_DS\_Report\_Team#

Solution

Assignment #4 (ASG4) Mon 10/4/21

Objective: Develop INSERT script from POPCO DGR (using the templated provided in D2L “DGR-Setup\_v2\_POPCO”)

1. Run DBDG\_v3 script in MSSQL
2. Run the following queries using DBDG\_v4a database in MSSQL
   1. How many databases do we have?
   2. How many tables each database has?
   3. For every table in SP database, identify number of attributes?
   4. list number of columns DGDB Database
   5. list table number, table name, and number of attributes in each table
   6. list database id, database name, table number, table name, column id, and column name
   7. List the following: Table\_ID, Table\_Name, column\_id, PK\_ID, constraint\_type, and constraint\_Name
   8. For every database, list tables without PK
   9. For every database, list tables without IDX
   10. For every database, list tables with PK but without FK
   11. For every database, list tables with PK but without IDX
   12. For every database, list tables with PK, FK, NN
   13. List the followings: database\_Name, number\_PKs, number\_FKs, number\_NNs, number\_CCs, number\_IDXs, number\_PGMs
   14. Develop a list to show how data type id used in every database
       1. Database name, Data\_Type, Count\_Data\_Type
   15. For every database, rank them according to number of indexes have created
3. Develop INSERT Statements for DGR-Setup\_v2\_POPCO document to be appended to the end of DGDB\_v4a and then save it as DBGD\_v4b
4. After adding Company DB, run the script (DBGD\_v4b ) to make sure it works
   1. Repeat all queries again using DBGD\_v4b

Deliverables:

1. A notepad to hold all new INSERTY Statments
2. A notepad document to hold all queries in English, SQL code, and result to the above questions
3. A completed spreadsheet for DGR\_Company INSERTs

2. One report per team ONLY include name of members that ARE present

3. Submit to ASG4 folder in D2L on due date

Solution

Deliverables: a word document to hold answers to the above questions

Submit to ASG4 folder in D2L by the end of class but no later than 11:59pm on the same day

SOLUTION

Assignment #3 ASG3) Fri 9/10/21

1. Using the spreadsheet provided for athletes participated in Olympics 2012, develop INSERT statements for the two work sheets (2012-Country\_medals-Genderand Player\_event-2012).
2. Past your INSERT Statements into a notepad and submit it to D2L
3. Based on using the Olympic data model, CREATE TABLE commands, and [Olympic\_data](#Olympic_Data) provided in two worksheets, develop INSERT statements.

Deliverables: a notepad document to hold answers to the above questions

Submit to ASG3 folder in D2L by the end of class but no later than 11:59pm on the same day

[SOLUTION](#ASG3_SOLUTION)

Assignment #2 (ASG2) Fri 9/3/21

Use the SP database for this assignment. Data model and script to create the database can be found in the public database folder.

* Your **report** should have query#, query description, SQL code, and result
* At least two of your queries should use subquery to join tables
* Use SQLPlus\* commands to format your report

Your task is to answer all the following questions.

1. List supplier number and supplier name for those suppliers that have status of 10 or 20
2. List part number and part name for those parts that their color is Red and their weight between 14 and 17
3. List part number and part name for those parts that have been shipped in quantity greater than 200
4. List supplier number and supplier name for those suppliers that have supplied red color parts
5. List supplier number, supplier name, and total quantity of all parts they have supplied
6. List part number and part name and their total quantity that have been shipped
7. Give supplier name that supplies the maximum quantity of parts.
8. Give name of the supplier whose supplies red parts and whose weight is greater than 10.
9. Give all part numbers that are from the same city. Make sure the result does not have any duplicate.
10. Give pair of supplier numbers that are from the same city
11. Give part (number and name) that has the minimum quantity
12. Give names of the suppliers who supplies the most parts and suppliers who supplies least parts
13. Give total quantity of all blue parts
14. List supplier number and supplier name that are not supplying any part.
15. Give top and bottom supplier number based on the quantity they supply.
16. Give top 3 parts (number and name) that weigh the least.
17. Give part number and part name that are not from London and whose color is not blue
18. Give name of suppliers that supply a red part.
19. Rank suppliers (supplier number and name) based on the total quantity of parts they supply.
20. List supplier number for the top 25% of suppliers in terms of total parts supplied

Deliverables: a notepad document to hold all your Insert statements

Submit to ASG2 folder in D2L by the end of class but no later than 11:59pm on the same day

SOLUTION

Assignment #1 (ASG1) Fri 8/27/21

Given these words: movie, theater, theater location, address, schedule, rating, , employee, show, Date, showtime, duration, and ticket price.

Your task is to: Create a well-formed DM using the information above to assist you.

* If the word is entity, then you can add appropriate attributes to it otherwise it could be used as an attribute.
* If your design establishes a weak entity, then you can add appropriate attributes to it

Goal: This exercise is intended to give you practice in forming DMs for a small database. You should take the time to look at each entity, attribute, relationship and identifier confirming that they are syntactically correct and that they mean what you want them to mean.

Grading Criteria:

* Neat, well-formed design, sensible decisions and assumption
* Drop a soft copy of your assignment in D2L.
* Late assignments will be penalized 25% per day
* No assignment will be accepted after it is solved and discussed in the class.

NOTE:

1. you need to draw the diagram using Visio or Lucide chart.
2. Make sure to state your assumptions very clearly for me to understand how you are viewing the design
3. Well-formed design
4. All entity name should be unique
5. All entity name should be singular
6. All entity name should be meaningful
7. Every entity should have identifier/primary key. Identifier could be single attribute or composite attributes
8. All attribute within one entity should have a unique name
9. All attribute within one entity should have a meaningful name
10. All attribute names should be singular
11. All attribute names should not be calculable
12. All attribute names should not be aggregate name (Address is an aggregate name. it should be broken down to street address, city, state, and zip code)
13. Relationship between two entities should be many-many. It should be one-one or many-one
14. More than one relationship between two entities should have label

Submit your solution to ASG1 folder in D2L by the end of class but no later than 11:59pm on the same day

SOLUTION

Zeota Data Model ([Back](#Phase1))

A picture containing diagram

Description automatically generated

Use\_Case template

Use\_Case ID:

Use\_Case description:

Use\_Case implementation plan in SQL :

Prepared by:

Verified by:

Date Created:

Date of last Update:

Sample of use-case template

Use\_Case ID:1

Use\_Case description: List of customers from every branch that have not ordered any product

Use\_Case implementation plan in SQL:

select b.branch\_no, c.Customer\_no, c.lname

from Branch B join Customer c on b.branch\_no=C.branch\_no

left join orders o on c.customer\_no=o.customer\_no

where o.customer\_no is null

Prepared by: John Black

Verified by: Adam Jones

Date Created: 9/13/21

Date of last Update:

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Screenshot of sample data from MDG document ([Back](#Inclass))

Table

Description automatically generated with low confidence

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**-- This script creates and populates the** **Company database in Oracle DBMS** ([Back](#Inclass))

-------------------------------------------------------

-- 1. Drop tables

DROP TABLE DEPARTMENT cascade constraint;

DROP TABLE EMPLOYEE cascade constraint;

DROP TABLE DEPENDENT cascade constraint;

DROP TABLE DEPT\_LOCATIONS cascade constraint;

DROP TABLE PROJECT cascade constraint;

DROP TABLE WORKS\_ON cascade constraint;

-------------------------------------------------------

-- 2. Create tables with PK and FK

CREATE TABLE DEPARTMENT(

DNUMBER NUMBER(2) NOT NULL,

DNAME VARCHAR(50) NOT NULL,

MGRSSN NUMBER(9)NOT NULL,

MGRSTARTDATE DATE NOT NULL,

CONSTRAINT DEPT\_DNUM\_PK PRIMARY KEY (DNUMBER));

CREATE TABLE EMPLOYEE(

SSN NUMBER (9) NOT NULL,

FNAME VARCHAR(25) NOT NULL,

MINIT VARCHAR(1),

LNAME VARCHAR(25) NOT NULL,

BDATE DATE NOT NULL,

ADDRESS VARCHAR(150) NOT NULL,

SEX VARCHAR(1) NOT NULL,

SALARY FLOAT,

SUPERSSN NUMBER(9),

DNO NUMBER(2) NOT NULL,

CONSTRAINT EMP\_SSN\_PK PRIMARY KEY(SSN),

CONSTRAINT EMP\_DNO\_FK FOREIGN KEY (DNO) REFERENCES DEPARTMENT(DNUMBER));

CREATE TABLE DEPENDENT(

ESSN NUMBER(9) NOT NULL,

DEPENDENT\_NAME VARCHAR(50) NOT NULL,

SEX VARCHAR(1) NOT NULL,

BDATE DATE NOT NULL,

RELATIONSHIP VARCHAR(15) NOT NULL,

CONSTRAINT DEP\_ESSN\_NAME\_PK PRIMARY KEY(ESSN, DEPENDENT\_NAME),

CONSTRAINT DEP\_ESSN\_FK FOREIGN KEY (ESSN) REFERENCES EMPLOYEE(SSN));

CREATE TABLE DEPT\_LOCATIONS(

DNUMBER NUMBER(2) NOT NULL,

DLOCATION VARCHAR(50) NOT NULL,

CONSTRAINT DEPTLOC\_DNUM\_DLOC\_PK PRIMARY KEY(DNUMBER, DLOCATION),

CONSTRAINT DEPTLOC\_DNUM\_FK FOREIGN KEY (DNUMBER) REFERENCES DEPARTMENT(DNUMBER));

CREATE TABLE PROJECT(

PNUMBER NUMBER(3) NOT NULL,

PNAME VARCHAR(25) NOT NULL,

PLOCATION VARCHAR(50),

DNUM NUMBER(2) NOT NULL,

CONSTRAINT PROJ\_PNUM\_PK PRIMARY KEY (PNUMBER) ,

CONSTRAINT PROJ\_DNUM\_FK FOREIGN KEY (DNUM) REFERENCES DEPARTMENT(DNUMBER));

CREATE TABLE WORKS\_ON(

ESSN NUMBER(9) NOT NULL,

PNO NUMBER(3) NOT NULL,

DNUM FLOAT,

CONSTRAINT WRK\_ESSN\_PNO\_PK PRIMARY KEY (ESSN, PNO),

CONSTRAINT WRK\_ESSN\_FK FOREIGN KEY (ESSN) REFERENCES EMPLOYEE(SSN),

CONSTRAINT WRK\_PNO\_FK FOREIGN KEY (PNO) REFERENCES PROJECT(PNUMBER));

-------------------------------------------------------

-- 3. Create indexes

--

-- 3.1. Drop indexes

DROP INDEX DEPTLOC\_DNUM\_IDX;

DROP INDEX PROJ\_DNUM\_IDX;

DROP INDEX EMP\_DNO\_IDX;

DROP INDEX DEP\_ESSN\_IDX;

DROP INDEX WRK\_ESSN\_IDX;

DROP INDEX WRK\_PNO\_IDX;

-- 3.2. Create indexes

CREATE INDEX DEPTLOC\_DNUM\_IDX ON DEPT\_LOCATIONS(DNUMBER);

CREATE INDEX PROJ\_DNUM\_IDX ON PROJECT(DNUM);

CREATE INDEX EMP\_DNO\_IDX ON EMPLOYEE(DNO);

CREATE INDEX DEP\_ESSN\_IDX ON DEPENDENT(ESSN);

CREATE INDEX WRK\_ESSN\_IDX ON WORKS\_ON(ESSN);

CREATE INDEX WRK\_PNO\_IDX ON WORKS\_ON(PNO);

-------------------------------------------------------

-- 4. grant privileges

grant select on DEPARTMENT to public;

grant select on EMPLOYEE to public;

grant select on DEPENDENT to public;

grant select on DEPT\_LOCATIONS to public;

grant select on PROJECT to public;

grant select on WORKS\_ON to public;

-------------------------------------------------------

-- 5. Insert values to SPJ's tables

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Company\_Worksheets ([Back](#Inclass))

Table

Description automatically generated with medium confidence

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Company Data Model (Company\_DM) ([Back](#Inclass))

Diagram

Description automatically generated

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ASG3\_SOLUTION [(back)](#DB_Admin)

\* Entering values into the 2012-Country\_medals-Gender table \*/

INSERT INTO 2012-Country\_medals\_Gender VALUES('USAUnited States ' , 17,15,13,45,'M') ;

INSERT INTO 2012-Country\_medals\_Gender VALUES('GBRGreat Britain ' , 17,9,13,39,'M') ;

INSERT INTO 2012-Country\_medals\_Gender VALUES('RUSRussia ' , 12,9,17,38, 'M') ;

INSERT INTO 2012-Country\_medals\_Gender VALUES('CHNChina ' , 17,8,11,36, 'M') ;

INSERT INTO 2012-Country\_medals\_Gender VALUES('GERGermany ' , 8,10,10,28, 'M') ;

...

/\* Insert statement for table Player\_event-2012 \*/

INSERT INTO Player\_event-2012 VALUES ('1001', 'Abdevali', ' Saeid', 'Islamic Republic of Iran', ' 11/04/1989', 'Wrestling' );

INSERT INTO Player\_event-2012 VALUES ('1002', 'Abdirahman', ' Abdi', 'United States of America', ' 01/01/1977', 'Track and Field' );

INSERT INTO Player\_event-2012 VALUES ('1003', 'Absalon', ' Julien', 'France', '08/16/1980', 'Cycling' );

INSERT INTO Player\_event-2012 VALUES ('1004', 'Acuff', ' Amy', 'United States of America', ' 07/14/1975', 'Track and Field' );

INSERT INTO Player\_event-2012 VALUES ('1005', 'Adams', ' Cammile', 'United States of America', ' 09/11/1991', 'Swimming' );

...

/\*NOTHER SOLUTION TO MANAGE DATE FIELD FOR PLAYER TABLE \*/

INSERT INTO Player\_event-2012 (Player\_ID, Last\_Name, First\_Name, COUNTRY, BORN, EVENT)

VALUES ( 1001, 'Abdevali', ' Saeid','Islamic Republic of Iran',TO\_DATE('04/11/1989','dd/mm/yyyy'), 'Wrestling' );

INSERT INTO Player\_event-2012 (Player\_ID, Last\_Name, First\_Name, COUNTRY, BORN,EVENT)

VALUES ( 1002, 'Abdirahman',' Abdi''United States of America',TO\_DATE('01/01/1977','dd/mm/yyyy'), 'Track and Field' );

INSERT INTO Player\_event-2012 (Player\_ID, Last\_Name, First\_Name, COUNTRY, BORN,EVENT)

VALUES ( 1003, 'Absalon',' Julien','France',TO\_DATE('16/08/1980','dd/mm/yyyy'), 'Cycling');

INSERT INTO Player\_event-2012 (Player\_ID, Last\_Name, First\_Name, COUNTRY, BORN,EVENT)

VALUES ( 1004, 'Acuff',' Amy','United States of America',TO\_DATE('14/07/1975','dd/mm/yyyy'), 'Track and Field' );

INSERT INTO Player\_event-2012 (Player\_ID, Last\_Name, First\_Name, COUNTRY, BORN,EVENT)

VALUES ( 1005, 'Adams',' Cammile', 'United States of America',TO\_DATE('11/09/1991','dd/mm/yyyy'), 'Swimming' );

....

Xxx

Olympic\_Data – Player [(back)](#DB_Admin)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Player\_ID | Last\_Name | First\_Name | COUNTRY | BORN | EVENT |
| 1001 | Abdevali | Saeid | Islamic Republic of Iran | 4-Nov-1989 | Wrestling |
| 1002 | Abdirahman | Abdi | United States of America | 1-Jan-1977 | Track and Field |
| 1003 | Absalon | Julien | France | 16-Aug-1980 | Cycling |
| 1004 | Acuff | Amy | United States of America | 14-Jul-1975 | Track and Field |
| 1005 | Adams | Cammile | United States of America | 11-Sep-1991 | Swimming |

…

Company data -country

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Nation | Gold Medal | Silver Medal | Bronze Medal | Total | Gender |
| USAUnited States | 17 | 15 | 13 | 45 | M |
| GBRGreat Britain | 17 | 9 | 13 | 39 | M |
| RUSRussia | 12 | 9 | 17 | 38 | M |
| CHNChina | 17 | 8 | 11 | 36 | M |
| GERGermany | 8 | 10 | 10 | 28 | M |
| KORSouth Korea | 8 | 7 | 6 | 21 | M |
| JPNJapan | 3 | 8 | 10 | 21 | M |

…