CSDA 5210 Ch. 5 SQL Practice

Please use the below scripts to build the ZAGI Retail Company Sales Dept DB and HAFH Realty Company Property Management DB in your Oracle Express Environment:


SQLCodeCreatePopulate_ZAGIMORE_O


SQLCodeCreatePopulate_HAFHMORE_C

E5.1 Write the SQL queries that accomplish the following tasks in the ZAGI Retail Company Sales Department Database:

- E5.1.1 Display all the records in the table REGION.
  SQL Command: select * from region;

REGIONID	REGIONNAME
C	Chicagoland
T	Tristate
I	Indiana
N	North

- E5.1.2 Display the StoreID and StoreZip for all stores.
  SQL Command: select StoreID, StoreZip from store;

STOREID	STOREZIP
S1	60600
S2	60605
S3	35400
S4	60640
S5	46307
S6	47374
S7	47401
S8	55401
S9	54937
S10	60602

More than 10 rows available. Increase rows selector to view more rows.

- E5.1.3 Display the CustomerName and CustomerZip for all customers, sorted alphabetically by CustomerName.
  SQL command: select customername, customerzip from customer order by customername;

CUSTOMERNAME	CUSTOMERZIP
Dan	55499
Elly	47374
Maggie	47401
Miles	60602
Neil	55403
Nora	60640
Pam	35401
Ryan	46202
Tina	60137
Tony	60611

- E5.1.4 Display the RegionIDs of regions where we have stores (use only table STORE and do not display the same information more than once).
  SQL Command: select distinct regionid from store;

REGIONID
I
C
T
N

- E5.1.5 Display all the information for all stores whose RegionID value is C.
  SQL Command: select * from store where regionid = 'C';

STOREID	STOREZIP	REGIONID
S1	60600	C
S2	60605	C
S4	60640	C
S10	60602	C

- E5.1.6 Display CustomerID and CustomerName for all customers whose CustomerName begins with a letter T.
  SQL Command: select customerid, customername from customer where customername like 'T%';

CUSTOMERID	CUSTOMERNAME
1-2-333	Tina
2-3-444	Tony

- E5.1.7 Display the ProductID, ProductName, and ProductPrice for products with a ProductPrice of $100 or higher.
  SQL Command: select productid, productname, productprice from product where productprice > '100';

PRODUCTID	PRODUCTNAME	PRODUCTPRICE
5X5	Tiny Tent	150
6X6	Biggy Tent	250
7X7	Hi-Tec GPS	300
1X2	Comfy Harness	150
1X3	Sunny Charger	125
4X3	Mega Camera	275
5X3	Luxo Tent	500

- E5.1.8 Display the ProductID, ProductName, ProductPrice, and VendorName for all products. Sort the results by ProductID.
  SQL Command: select p.productid, p.productname, p.productprice, v.vendorname from product p, vendor v where p.vendorid = v.vendorid order by productid;

PRODUCTID	PRODUCTNAME	PRODUCTPRICE	VENDORNAME
1X1	Zzz Bag	100	Pacifica Gear
1X2	Comfy Harness	150	Mountain King
1X3	Sunny Charger	125	Outdoor Adventures
1X4	Safe-T Helmet	40	Pacifica Gear
2X1	Mmm Stove	80	Wilderness Limited
2X2	Easy Boot	70	Mountain King
2X3	Reflect-o Jacket	35	Pacifica Gear
2X4	Strongster Carribeaner	20	Mountain King
3X1	Sleepy Pad	25	Wilderness Limited
3X2	Bucky Knife	60	Wilderness Limited

- E5.1.9 Display the ProductID, ProductName, ProductPrice, VendorName, and CategoryName for all products. Sort the results by ProductID.
  SQL Command: select p.productid, p.productname, p.productprice, v.vendorname, c.categoryname from product p, vendor v, category c where p.vendorid = v.vendorid and p.categoryid = c.categoryid order by productid;

PRODUCTID	PRODUCTNAME	PRODUCTPRICE	VENDORNAME	CATEGORYNAME
1X1	Zzz Bag	100	Pacifica Gear	Camping
1X2	Comfy Harness	150	Mountain King	Climbing
1X3	Sunny Charger	125	Outdoor Adventures	Electronics
1X4	Safe-T Helmet	40	Pacifica Gear	Cycling
2X1	Mmm Stove	80	Wilderness Limited	Camping
2X2	Easy Boot	70	Mountain King	Footwear
2X3	Reflect-o Jacket	35	Pacifica Gear	Cycling
2X4	Strongster Carribeaner	20	Mountain King	Climbing
3X1	Sleepy Pad	25	Wilderness Limited	Camping
3X2	Bucky Knife	60	Wilderness Limited	Camping

- E5.1.10 Display the ProductID, ProductName, and ProductPrice for products in the category whose CategoryName value is Camping. Sort the results by ProductID.
  SQL Command: select p.productid, p.productname, p.productprice from product p, category c where p.categoryid = c.categoryid and c.categoryname = 'Camping' order by p.productid;

PRODUCTID	PRODUCTNAME	PRODUCTPRICE
1X1	Zzz Bag	100
2X1	Mmm Stove	80
3X1	Sleepy Pad	25
3X2	Bucky Knife	60
5X3	Luxo Tent	500
5X5	Tiny Tent	150
6X6	Biggy Tent	250

7 rows returned in 0.00 seconds Download

- E.5.1.11 Display the ProductID, ProductName, and ProductPrice for products that were sold in the zip code 60600. Sort the results by ProductID.
  SQL Command: select p.productid, p.productname, p.productprice from product p, includes i, salestransaction t, store s where p.productid = i.productid and t.tid = i.tid and s.storeid = t.storeid and s.storezip = '60600' order by p.productid;

PRODUCTID	PRODUCTNAME	PRODUCTPRICE
1X1	Zzz Bag	100

1 rows returned in 0.00 seconds Download

- E.5.1.12 Display the ProductID, ProductName, and ProductPrice for Products whose VendorName is Pacifica Gear that were sold in the region whose RegionName is Tristate. Do not display the same information more than once (i.e., do not show the same product more than once). Sort the results by ProductID.
  SQL Command: select distinct p.productid, p.productname, p.productprice from product p, vendor v, includes i, salestransaction t, store s, region r where p.vendorid = v.vendorid and p.productid = i.productid and t.tid = i.tid and s.storeid = t.storeid and s.regionid = r.regionid and r.regionname = 'Tristate' and v.vendorname = 'Pacifica Gear' order by p.productid;

PRODUCTID	PRODUCTNAME	PRODUCTPRICE
1X1	Zzz Bag	100
4X4	Dura Boot	90
5X1	Simple Sandal	50

3 rows returned in 0.13 seconds Download

- E5.1.13 Display the TID, CustomerName, and TDate for sales transactions involving a customer buying a product whose ProductName is Easy Boot.
  SQL Command: select t.tid, c.customername, t.tdate from salestransaction t, customer c, product p, includes I where t.tid = i.tid and p.productid = i.productid and c.customerid = t.customerid and p.productname = 'Easy Boot';

TID	CUSTOMERNAME	TDATE
T022	Ryan	01/07/2020
T222	Tony	01/01/2020
T444	Pam	01/02/2020
T505	Miles	01/05/2020
T606	Dan	01/06/2020
T808	Neil	01/06/2020

6 rows returned in 0.02 seconds Download

- E5.1.14 Display the RegionID, RegionName, and number of stores in the region for all regions.
  SQL Command: select r.regionid, r.regionname, count(*) from region r, store s where r.regionid = s.regionid group by r.regionid, r.regionname;

REGIONID	REGIONNAME	COUNT(*)
C	Chicagoland	4
I	Indiana	3
N	North	3
T	Tristate	4

- E5.1.15 For each product category, display the CategoryID, CategoryName, and average price of a product in the category.
  SQL Command: select c.categoryid, c.categoryname, avg(p.productprice) from product p, category c where p.categoryid = c.categoryid group by c.categoryid, c.categoryname;

CATEGORYID	CATEGORYNAME	AVG(P.PRODUCTPRICE)
CL	Climbing	66.666666666666666666666666666666666667
CP	Camping	166.42857142857142857142857142857142857571428571
CY	Cycling	30
EL	Electronics	186.25
FW	Footwear	59

5 rows returned in 0.02 seconds Download

- E5.1.16 For each product category, display the CategoryID and the total number of items purchased in the category.
  SQL Command: select p.categoryid, sum(i.quantity) from includes i, product p where i.productid = p.productid group by p.categoryid;

CATEGORYID	SUM(I.QUANTITY)
EL	12
CY	24
CP	21
CL	25
FW	34

5 rows returned in 0.00 seconds Download

- E5.1.17 Display the RegionID, RegionName, and the total amount of sales (in dollars) in the region for all regions. Display the total amount of sales as AmountSpent.

SQL Command: select r.regionid, r.regionname, sum(i.quantity*p.productprice) as amountspent from includes i, product p, salestransaction t, store s, region r where i.productid = p.productid and t.tid = i.tid and t.storeid = s.storeid and s.regionid = r.regionid group by r.regionid, r.regionname;

REGIONID	REGIONNAME	AMOUNTSPENT
C	Chicagoland	1630
I	Indiana	2170
N	North	1520
T	Tristate	3060

4 rows returned in 0.34 seconds Download

- E5.1.18 Display the TID and the total number of items (of all products) sold within the transaction for all sales transactions whose total number of items (of all products) sold within the transaction is greater than five.
  SQL Command: select tid, sum(quantity) from includes group by tid having sum(quantity) > 5;

TID	SUM(QUANTITY)
T707	7
T888	7
T555	7
T999	10
T505	8
T606	18
T022	8
T333	6
T303	9
T808	8

10 rows returned in 0.01 seconds Download

- E5.1.19 For each vendor whose product sales exceed $700, display the VendorID, VendorName, and total amount of sales in dollars. Display the total amount of sales as TotalSales.
  SQL Command: select v.vendorid, v.vendorname, sum(i.quantity*p.productprice) as totalsales from includes i, product p, vendor v where i.productid = p.productid and p.vendorid = v.vendorid group by v.vendorid, v.vendorname having sum(i.quantity*p.productprice) > 700;

VENDORID	VENDORNAME	TOTALSALES
OA	Outdoor Adventures	1405
MK	Mountain King	3590
PG	Pacifica Gear	1385
WL	Wilderness Limited	2000

4 rows returned in 0.00 seconds Download

- E5.1.20 Display the ProductID, ProductName, and ProductPrice of the cheapest product.
  SQL Command: select productid, productname, productprice from product where productprice = (select min(productprice) from product);

PRODUCTID	PRODUCTNAME	PRODUCTPRICE
3X3	Cosy Sock	15

- E5.1.21 Display the ProductID, ProductName, and VendorName for products whose price is below the average price of all products.
  SQL Command: select p.productid, p.productname, v.vendorname from product p, vendor v where p.vendorid = v.vendorid and productprice < (select avg(productprice) from product);

PRODUCTID	PRODUCTNAME	VENDORNAME
1X1	Zzz Bag	Pacifica Gear
2X2	Easy Boot	Mountain King
3X3	Cosy Sock	Mountain King
4X4	Dura Boot	Pacifica Gear
8X8	Power Pedals	Mountain King
9X9	Trusty Rope	Wilderness Limited
1X4	Safe-T Helmet	Pacifica Gear
2X1	Mmm Stove	Wilderness Limited
2X3	Reflect-o Jacket	Pacifica Gear
2X4	Strongster Carribeaner	Mountain King

- E5.1.22 Display the ProductID and ProductName of the product for the products whose total quantity sold in all transactions is greater than 2. Sort the results by ProductID.
  SQL Command: select p.productid, p.productname from product p, includes I where i.productid = p.productid group by p.productid, p.productname having sum(i.quantity) > 2 order by productid;

PRODUCTID	PRODUCTNAME
1X2	Comfy Harness
1X3	Sunny Charger
1X4	Safe-T Helmet
2X1	Mmm Stove
2X2	Easy Boot
2X3	Reflect-o Jacket
2X4	Strongster Carribeaner
3X1	Sleepy Pad
3X3	Cosy Sock
3X4	Treado Tire

More than 10 rows available. Increase rows selector to view more rows.

- E5.1.23 Display the ProductID for the product that has been sold the most within all transactions (i.e., that has been sold in the highest total quantity across all transactions).
  SQL Command: select productid from includes group by productid having sum(quantity) = (select max(A) from (select sum(quantity) as A from includes group by productid));

PRODUCTID
2X2

1 rows returned in 0.00 seconds Download

- E5.1.24 Rewrite Query 30 using a join statement (no nested queries).

Query 30 text: *For each product that has more than three items sold within all sales transactions, retrieve the product id, product name, and product price.*

Query 30:
```
SELECT productid, productname, productprice
        FROM product
        WHERE productid IN
             (SELECT productid
             FROM includes
             GROUP BY productid
             HAVING SUM(quantity) > 3);
```

SQL Command: select p.productid, p.productname, p.productprice from product p, includes s where p.productid = s.productid group by p.productid, p.productname, p.productprice having sum(s.quantity) > 3;

PRODUCTID	PRODUCTNAME	PRODUCTPRICE
2X3	Reflect-o Jacket	35
1X2	Comfy Harness	150
2X4	Strongster Carribeaner	20
8X8	Power Pedals	20
4X3	Mega Camera	275
4X2	Electra Compass	45
3X4	Treado Tire	30
9X9	Trusty Rope	30
1X4	Safe-T Helmet	40
2X2	Easy Boot	70

More than 10 rows available. Increase rows selector to view more rows.

- E5.1.25 Rewrite Query 31 using a join statement (no nested queries).

Query 31 text: *For each product whose items were sold in more than one sales transaction, retrieve the product id, product name, and product price.*

Query 31:
```
SELECT productid, productname, productprice
        FROM product
        WHERE productid IN
              (SELECT productid
              FROM includes
              GROUP BY productid
              HAVING COUNT (tid) > 1);
```

SQL Command: select p.productid, p.productname, p.productprice from product p, includes s where p.productid = s.productid group by p.productid, p.productname, p.productprice having count(s.tid) > 1;

PRODUCTID	PRODUCTNAME	PRODUCTPRICE
2X3	Reflect-o Jacket	35
1X2	Comfy Harness	150
6X6	Biggy Tent	250
1X3	Sunny Charger	125
2X4	Strongster Carribeaner	20
8X8	Power Pedals	20
4X3	Mega Camera	275
4X2	Electra Compass	45
1X1	Zzz Bag	100
3X4	Treado Tire	30

More than 10 rows available. Increase rows selector to view more rows.

E5.2 Write the SQL queries that accomplish the following tasks in the HAFH Realty Company Property Management Database:

- E5.2.1 Display all the records in the table STAFFMEMBER.
  SQL Command: select * from staffmember;

SMEMBERID	SMEMBERNAME
5432	Brian
9876	Boris
7652	Caroline
2537	Howard
3984	Luis
4196	Arthur
8467	Mariana
1028	Franz

8 rows returned in 0.01 seconds Download

- E5.2.2 Display the BuildingID and number of floors in the building for all buildings.
  SQL Command: select buildingid, bnooffloors from building;

BUILDINGID	BNOOFFLOORS
B1	5
B2	6
B3	4
B4	4
B5	3
B6	3
B7	2
B8	4
B9	3

- E5.2.3 Display the CCID, CCName, and CCIndustry for all corporate clients, sorted alphabetically by CCName.
  SQL Command: select ccid, ccname, ccindustry from corpclient order by ccname;

CCID	CCNAME	CCINDUSTRY
C111	BlingNotes	Music
C999	CommuteAir	Airline
C666	DelishInc	Food Service
C555	EntertainUs	Broadcasting
C444	NanoCorp	Broadcasting
C222	SkyJet	Airline
C888	SouthAlps	Sports
C777	WindyCT	Music
C333	Xilerate	Sports

- E5.2.4 Display all the records in the table STAFFMEMBER for staff members whose SMemberName begins with a letter B.
  SQL Command: select * from staffmember where smembername like 'B%';

SMEMBERID	SMEMBERNAME
5432	Brian
9876	Boris

- E5.2.5 Display the BuildingID, AptNo, and ANoOfBedrooms for all apartments that have more than one bedroom.
  SQL Command: select buildingid, aptno, anoofbedrooms from apartment where anoofbedrooms > 1;

BUILDINGID	APTNO	ANOOFBEDROOMS
B2	11	2
B2	21	2
B2	31	2
B2	41	2
B2	51	2
B2	61	2
B3	11	2
B3	21	2
B3	31	2
B3	41	2

More than 10 rows available. Increase rows selector to view more rows.

- E5.2.6 Display the answer to the following question: How many HAFH buildings have exactly 4 floors?
  SQL Command: select count(*) from building where bnooffloors = 4;

COUNT(*)
3

- E5.2.7 Display the total amount HAFH spends on manager salaries (as TotalSalary) and the total amount HAFH spends on manager bonuses (as TotalBonus).
  SQL Command: select sum(msalary) as totalsalary, sum(mbonus) as totalbonus from manager;

TOTALSALARY	TOTALBONUS
335000	12000

- E5.2.8 Display the ManagerID, MFName, MLName, MSalary, and MBonus, for all managers with a salary greater than $50,000 and a bonus greater than $1,000.
  SQL Command: select managerid, mfname, mlname, msalary, mbonus from manager where msalary > 50000 and mbonus > 1000;

MANAGERID	MFNAME	MLNAME	MSALARY	MBONUS
M34	George	Sherman	52000	2000
M56	Fiona	Keane	57000	2000
M67	Alexander	Sanborn	62000	3000

- E5.2.9 Display the BuildingID, BNoOfFloors, and the manager's MFName and MLName for all buildings.
  SQL Command: select b.buildingid, b.bnooffloors, m.mfname, m.mlname from building b, manager m where b.bmanagerid = m.managerid;

BUILDINGID	BNOOFFLOORS	MFNAME	MLNAME
B1	5	Boris	Grant
B2	6	Austin	Lee
B3	4	Austin	Lee
B4	4	George	Sherman
B5	3	Mariana	Gonzalez
B6	3	Mariana	Gonzalez
B7	2	Fiona	Keane
B8	4	Alexander	Sanborn
B9	3	Alexander	Sanborn

- E5.2.10 Display the MFName, MLName, MSalary, MBdate, and number of buildings that the manager manages for all managers with a salary less than $55,000.
  SQL Command: select m.mfname, m.mlname, m.msalary, m.mbdate, count(*) from building b, manager m where b.bmanagerid = m.managerid AND msalary < 55000 group by m.mfname, m.mlname, m.msalary, mbdate;

MFNAME	MLNAME	MSALARY	MBDATE	COUNT(*)
Mariana	Gonzalez	54000	12/27/1980	2
Austin	Lee	50000	10/30/1975	2
George	Sherman	52000	01/11/1976	1

3 rows returned in 0.03 seconds Download

- E5.2.11 Display the BuildingID and AptNo for all apartments leased by the corporate client WindyCT.
  SQL Command: select a.buildingid, a.aptno from apartment a, corpclient c where a.ccid = c.ccid AND c.ccname = 'WindyCT';

BUILDINGID	APTNO
B3	11
B3	21
B4	11
B4	21
B8	11
B8	12

6 rows returned in 0.00 seconds Download

- E5.2.12 Display the InsID and InsName for all inspectors that have any inspections scheduled after 1-APR-2020. Do not display the same information more than once.
  SQL Command: select distinct ir.insid, ir.insname from inspector ir, inspecting ig where ir.insid = ig.insid AND ig.datenext > '04-01-2020';

INSID	INSNAME
I11	Jane
I33	Mick
I44	Bianca
I55	Sergei

4 rows returned in 0.03 seconds Download

- E5.2.13 Display the ManagerID, MFName, MLName, and MSalary, for all managers with a salary greater than the average manager's salary.
  SQL Command: select managerid, mfname, mlname, msalary from manager where msalary > (select avg(msalary) from manager);

MANAGERID	MFNAME	MLNAME	MSALARY
M12	Boris	Grant	60000
M56	Fiona	Keane	57000
M67	Alexander	Sanborn	62000

3 rows returned in 0.00 seconds Download

- E5.2.14 Display the ManagerID, MFName, MLName, and the number of buildings managed, for all managers.
  SQL Command: select m.managerid, m.mfname, m.mlname, count(*) from manager m, building b where b.bmanagerid = m.managerid group by m.managerid, m.mfname, m.mlname;

MANAGERID	MFNAME	MLNAME	COUNT(*)
M56	Fiona	Keane	1
M67	Alexander	Sanborn	2
M45	Mariana	Gonzalez	2
M23	Austin	Lee	2
M12	Boris	Grant	1
M34	George	Sherman	1

6 rows returned in 0.01 seconds Download

- E5.2.15 Display the ManagerID, MFName, MLName, and number of buildings managed, for all managers that manage more than one building.
  SQL Command: select m.managerid, m.mfname, m.mlname, count(*) from manager m, building b where b.bmanagerid = m.managerid group by m.managerid, m.mfname, m.mlname having count(*) > 1;

MANAGERID	MFNAME	MLNAME	COUNT(*)
M67	Alexander	Sanborn	2
M45	Mariana	Gonzalez	2
M23	Austin	Lee	2

3 rows returned in 0.02 seconds Download

- E5.2.16 Display the SMemberID, SMemberName, and the number of apartments that the staff members cleans, for all staff members.
  SQL Command: select s.smemberid, s.smembername, count(*) from staffmember s, cleaning c where s.smemberid = c.smemberid group by s.smemberid, s.smembername;

SMEMBERID	SMEMBERNAME	COUNT(*)
1028	Franz	8
2537	Howard	4
3984	Luis	3
4196	Arthur	6
5432	Brian	7
7652	Caroline	6
8467	Mariana	5
9876	Boris	6

8 rows returned in 0.00 seconds Download

- E5.2.17 Display the SMemberID and SMemberName of staff members cleaning apartments rented by corporate clients whose corporate location is Chicago. Do not display the same information more than once.
  SQL Command: select distinct s.smemberid, s.smembername from staffmember s, cleaning cl, apartment a, corpclient cc where s.smemberid = cl.smemberid and a.aptno = cl.aptno and a.buildingid = cl.buildingid and a.ccid = cc.ccid and cc.cclocation = 'Chicago';

SMEMBERID	SMEMBERNAME
1028	Franz
2537	Howard
7652	Caroline
8467	Mariana
5432	Brian
4196	Arthur
3984	Luis

7 rows returned in 0.06 seconds Download

- E5.2.18 Display the CCName of the client and the CCName of the client who referred him or her, for every client referred by a client in the Music industry.
  SQL Command: select c1.ccname client, c2.ccname referredby from corpclient c1, corpclient c2 where c2.ccid = c1.ccidreferredby and c2.ccindustry = 'Music';

CLIENT	REFERREDBY
SkyJet	BlingNotes
NanoCorp	BlingNotes
CommuteAir	BlingNotes
SouthAlps	WindyCT

4 rows returned in 0.02 seconds Download

- E5.2.19 Display the BuildingID, AptNo, and ANoOfBedrooms for all apartments that are not leased.
  SQL Command: select buildingid, aptno, anoofbedrooms from apartment where ccid is null;

BUILDINGID	APTNO	ANOOFBEDROOMS
B1	41	1
B1	51	1
B2	31	2
B2	41	2
B5	21	3
B7	21	3
B8	31	2
B8	32	2

8 rows returned in 0.00 seconds Download

- E5.2.20 Display the ManagerID, MFName, MLName, MSalary, and MBonus, for the manager with the lowest total compensation (total compensation is defined as the salary plus bonus).

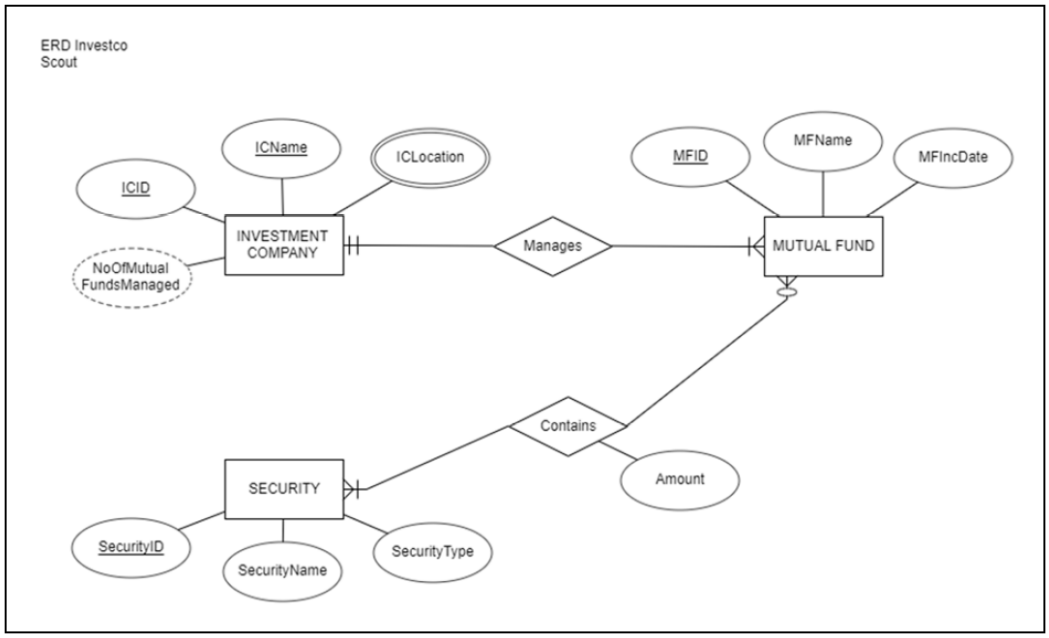
SQL Command: select managerid, mfname, mlname, msalary, COALESCE(mbonus,0) as mgrbonus from manager where msalary+COALESCE(mbonus,0) = (select min(msalary+COALESCE(mbonus,0)) from manager);

MANAGERID	MFNAME	MLNAME	MSALARY	MGRBONUS
M34	George	Sherman	52000	2000
M45	Mariana	Gonzalez	54000	0

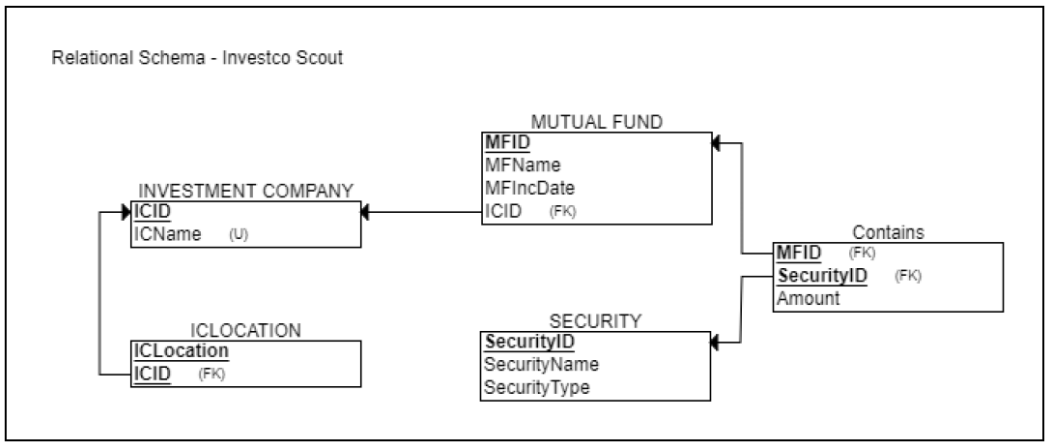
2 rows returned in 0.02 seconds Download

Mini Case Invesco Scout

ERD:



Relational Schema:



Write CREATE TABLE statements to create the tables for the Investco Scout Funds Database depicted by the relational schema

Assume that the following addition is made to the Investco Scout Funds Database:

> *Investco Scout will keep track of the CEOFName and CEOLName for each investment company (in addition to keeping track of a unique investment company identifier, a unique investment company name, and names of multiple locations of the investment company for each investment company).*

Change the ER diagram new addition to the requirements.

Change the relational schema to reflect the change in the ER diagram.

Write ALTER TABLE commands that reflect the change in the relational schema.

Observe the following information about investment companies and mutual funds, and create INSERT INTO statements that insert the information into the created tables of the Investco Scout Funds Database.

Investment Companies		
Company: ACF, Acme Finance	CEO: Mick Dempsey	Locations: Chicago, Denver
Company: TCA, Tara Capital	CEO: Ava Newton	Locations: Houston, New York City
Company: ALB, Albritton	CEO: Lena Dollar	Locations: Atlanta, New York City

Securities (ID, Name, Type)		
AE	Abhi Engineering	Stock
BH	Blues Health	Stock
CM	County Municipality	Bond
DU	Downtown Utility	Bond
EM	Emmitt Machines	Stock

Mutual Funds by Investment Company (Inception Date, ID, Name, Mix)			
ACF:			
1/1/2005	BG	Big Growth	(500 AE Stocks, 300 EM Stocks)
1/1/2006	SG	Steady Growth	(300 AE Stocks, 300 DU Bonds)
TCA:			
1/1/2005	LF	Tiger Fund	(1000 EM Stocks, 1000 BH Stocks)
1/1/2006	OF	Owl Fund	(1000 CM Bonds, 1000 DU Bonds)
ALB:			
1/1/2005	JU	Jupiter	(2000 EM Stock, 1000 DU Bonds)
1/1/2006	SA	Saturn	(1000 EM Stock, 2000 DU Bonds)

Create table statements:

```
CREATE TABLE investmentcompany
(
icid CHAR(4) NOT NULL,
icname VARCHAR NOT NULL,
PRIMARY KEY (icid),
UNIQUE (icname)
);

CREATE TABLE mutualfund
(
mfid CHAR(4) NOT NULL,
mfname VARCHAR NOT NULL,
mfincdate DATE NOT NULL,
icid CHAR(4) NOT NULL,
PRIMARY KEY (mfid),
FOREIGN KEY (icid) REFERENCES investmentcompany(icid)
);

CREATE TABLE security
(
securityid CHAR(3) NOT NULL,
securityname VARCHAR NOT NULL,
securitytype VARCHAR NOT NULL,
PRIMARY KEY (securityid)
);

CREATE TABLE contains
(
mfid CHAR(4) NOT NULL,
```

```
securityid CHAR(3) NOT NULL,
amount INT NOT NULL,
PRIMARY KEY (mfid, securityid),
FOREIGN KEY (mfid) REFERENCES mutualfund(mfid),
FOREIGN KEY (securityid) REFERENCES security(securityid)
);

CREATE TABLE iclocation
(
icid CHAR(4) NOT NULL,
iclocation VARCHAR NOT NULL,
PRIMARY KEY (iclocation, icid),
FOREIGN KEY (icid) REFERENCES investmentcompany(icid)
);
```

Insert table statements:

```
INSERT INTO investmentcompany VALUES ('ACF', 'Acme Finance', 'Mick', 'Dempsey');
INSERT INTO investmentcompany VALUES ('TCA', 'Tara Capital', 'Ava', 'Newton');
INSERT INTO investmentcompany VALUES ('ALB', 'Albritton', 'Lena', 'Dollar');
INSERT INTO mutualfund VALUES ('BG', ' Big Growth', '1/Jan/2005', 'ACF');
INSERT INTO mutualfund VALUES ('SG', ' Steady Growth', '1/Jan/2006', 'ACF');
INSERT INTO mutualfund VALUES ('LF', ' Tiger Fund', '1/Jan/2005', 'TCA');
INSERT INTO mutualfund VALUES ('OF', ' Owl Fund', '1/Jan/2006', 'TCA');
INSERT INTO mutualfund VALUES ('JU', ' Jupiter', '1/Jan/2005', 'ALB');
INSERT INTO mutualfund VALUES ('SA', ' Saturn', '1/Jan/2006', 'ALB');
INSERT INTO security VALUES ('AE', 'Abhi Engineering', 'Stock');
INSERT INTO security VALUES ('BH', 'Blues Health', 'Stock');
INSERT INTO security VALUES ('CM', 'County Municipality', 'Bond');
```

```
INSERT INTO security VALUES ('DU', 'Downtown Utility', 'Bond');
INSERT INTO security VALUES ('EM', 'Emmitt Machines', 'Stock');
INSERT INTO contains VALUES ('BG', ' AE', 500);
INSERT INTO contains VALUES ('BG', ' EM', 300);
INSERT INTO contains VALUES ('SG', ' AE', 300);
INSERT INTO contains VALUES ('SG', ' DU', 300);
INSERT INTO contains VALUES ('LF', ' EM', 1000);
INSERT INTO contains VALUES ('LF', ' BH', 1000);
INSERT INTO contains VALUES ('OF', ' CM', 1000);
INSERT INTO contains VALUES ('OF', ' DU', 1000);
INSERT INTO contains VALUES ('JU', ' EM', 2000);
INSERT INTO contains VALUES ('JU', ' DU', 1000);
INSERT INTO contains VALUES ('SA', ' EM', 1000);
INSERT INTO contains VALUES ('SA', ' DU', 2000);
INSERT INTO iclocation VALUES ('ACF', 'Chicago');
INSERT INTO iclocation VALUES ('ACF', 'Denver');
INSERT INTO iclocation VALUES ('TCA', 'Houston');
INSERT INTO iclocation VALUES ('TCA', 'New York City');
INSERT INTO iclocation VALUES ('ALB', 'Atlanta');
INSERT INTO iclocation VALUES ('ALB', 'New York City');
```