



Bahria University
Discovering Knowledge

Department of Computer Science
Database Management System
Semester 4 Section A (fall 2022)

Project Name: Mart Management System

Analyzed Application URL:

1. <https://www.shopify.com/>
2. <https://www.daraz.pk/>
3. <https://www.hummart.com/>

Class: BSCS - 4(A)

Group Members:

Student Name	Enrollment	Viva Marks
Ashir Azeem	02-134211-039	
Muhammad Huzaifa	02-134211-059	
Aqsa Malik	02-1342111-106	

Project Marks:

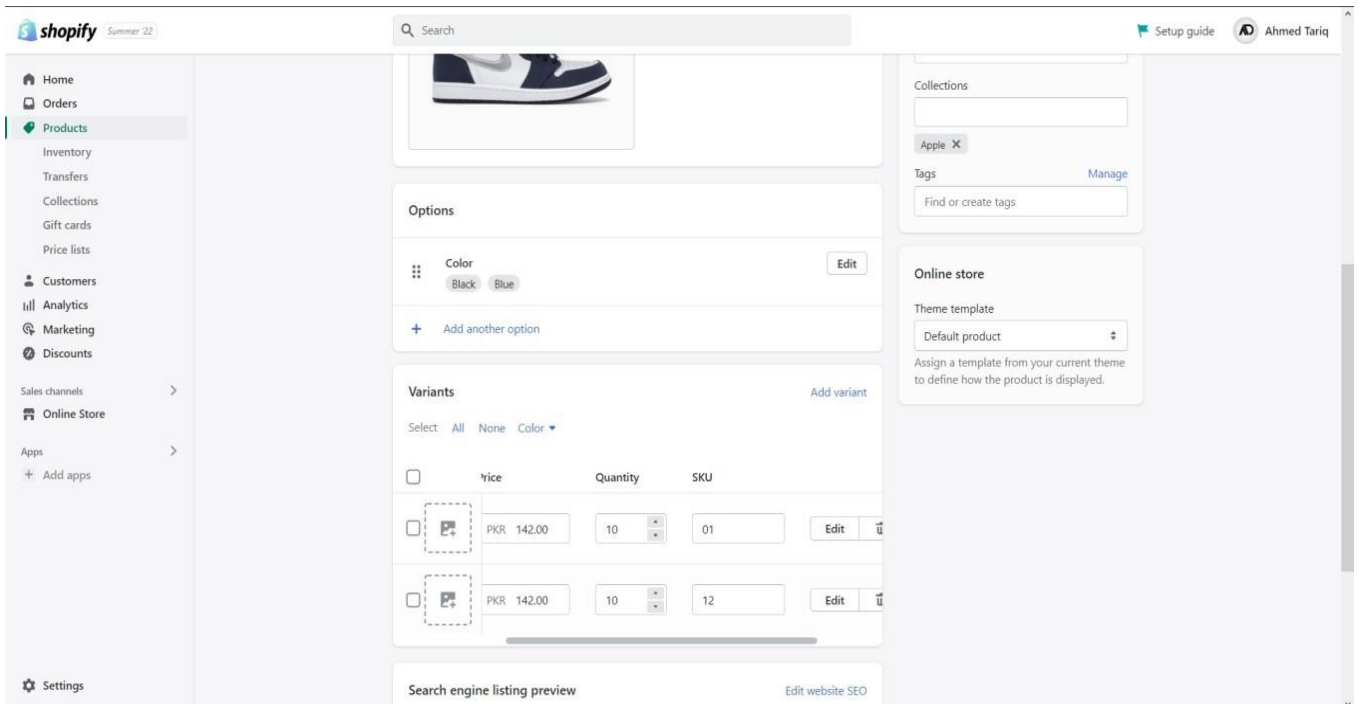
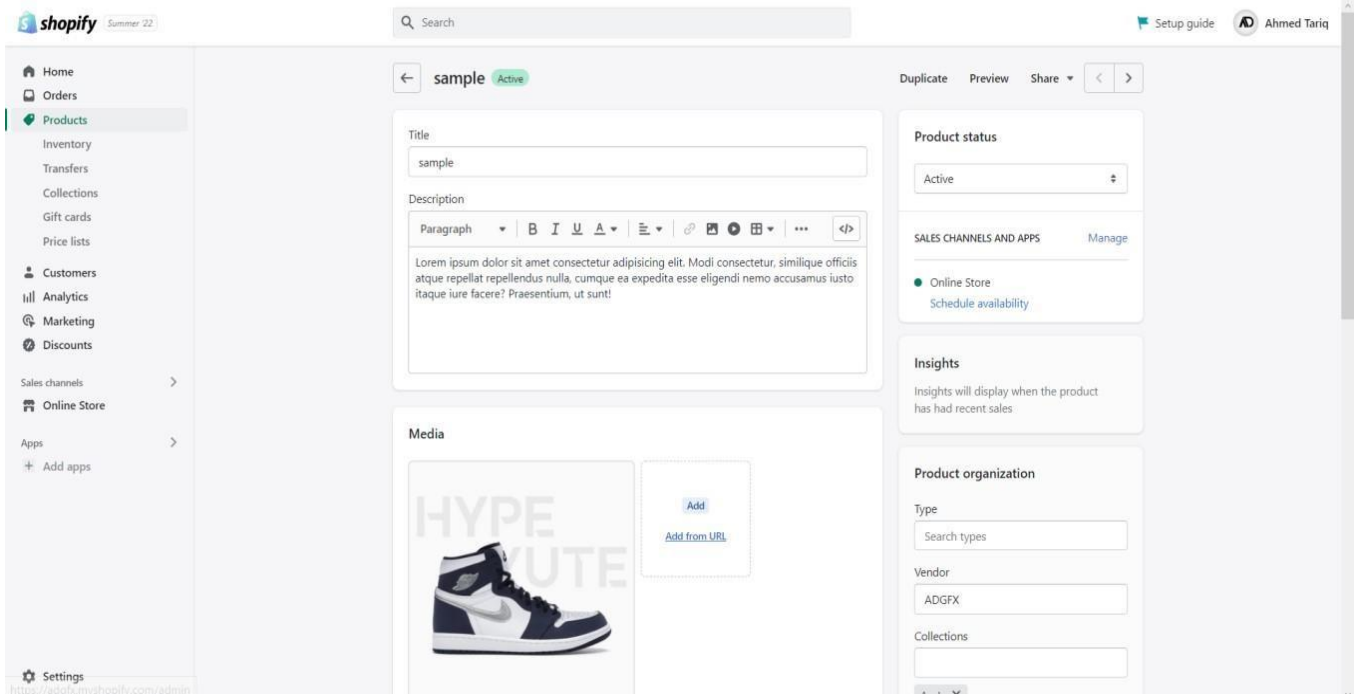
Head	Performance	Comments
Analysis & Report		
ERD		
Normalization		
DDL/DML/Triggers		
Stored Proc/ Views/ Stored Functions		

1)Analysis – Screenshots of the Application :

HUMMART:



SHOPIFY:



The screenshot displays the Shopify Admin interface for the 'adgfh.myshopify.com' store. The left sidebar shows the navigation menu with 'Products' highlighted. The main content area is the 'Inventory' page, which lists 12 inventory items. The table has the following columns: Product, SKU, Incoming, Committed, and Available. The items listed are:

Product	SKU	Incoming	Committed	Available
Black	No SKU	0	0	10
2 White	No SKU	0	0	0
3 Black	No SKU	0	0	10
3 Blue	No SKU	0	0	10
4 Black	No SKU	0	0	10
4 Blue	No SKU	0	0	10
5 Black	No SKU	0	0	10
5 Blue	No SKU	0	0	10
6 Black	No SKU	0	0	10
6 Blue	No SKU	0	0	10
sample Black	01	0	0	10
sample Blue	12	0	0	10

At the bottom of the page, there is a link: [Learn more about managing inventory.](#)

Stores

Referrals

Affiliate tools

Apps

Themes

Payouts

Shopify POS

Partner Resources

Support

Team

Settings

Unsaved store

< Stores

Add store

Store type

Each store type has unique limitations and features designed for its particular purpose. Pick a store type that best suits your needs.

You can test how customers pay for their orders using the [Shopify Bogus Gateway or Shopify Payments test mode](#).

You can give staff members the ability to work on this store after it's been added.

Store name

Store URL

Login

We'll use this address if we need to contact you about your account.

Password

Confirm password

DARAZ:



2) Business Rules for the DBMS:

STORE:

- Each store has a unique identification number, There are several departments in one store, each department has a unique identification number
- Departments can be further classified as ADMIN department, HR department and FINANCE department each of these departments have its unique department identification number
- Multiple employees are working under these departments. One employee is assigned to 1 department.

- Each of the employee has its unique identification number.
- Payroll of employees may change due to different factors. Each payroll has unique id and against each id, one employee is assigned.
- One employee may be assigned to different roles and one role may be given to more than 1 employee.
- Each employee is given a unique login id. One login-id will be assigned to only one particular employee.
- Each employee will have a unique location id.

ADMIN:

- One admin can add multiple products and each product will be added by 1 admin only.
- One admin can initiate multiple promotions and each promotion will be initiated by 1 admin only.
- Multiple products can belong to 1 category and each product will have only one category.
- Each product with unique identification number will have multiple stocks.
- Multiple products may belong to a unique brand having unique brand id.
- One or more products may be supplied by a supplier having a unique supplier id. one supplier may supply multiple products.
- One or more products may be included in promotion having unique promotion id and one promotion may include multiple products.
- One or more products may be included in one order with a unique order identification number and one order may include different and multiple products.

HR:

- Department manages multiple feedbacks. Each feedback will be having a unique feedback id.
- Department conducts multiple appraisals with each appraisal having a unique id.

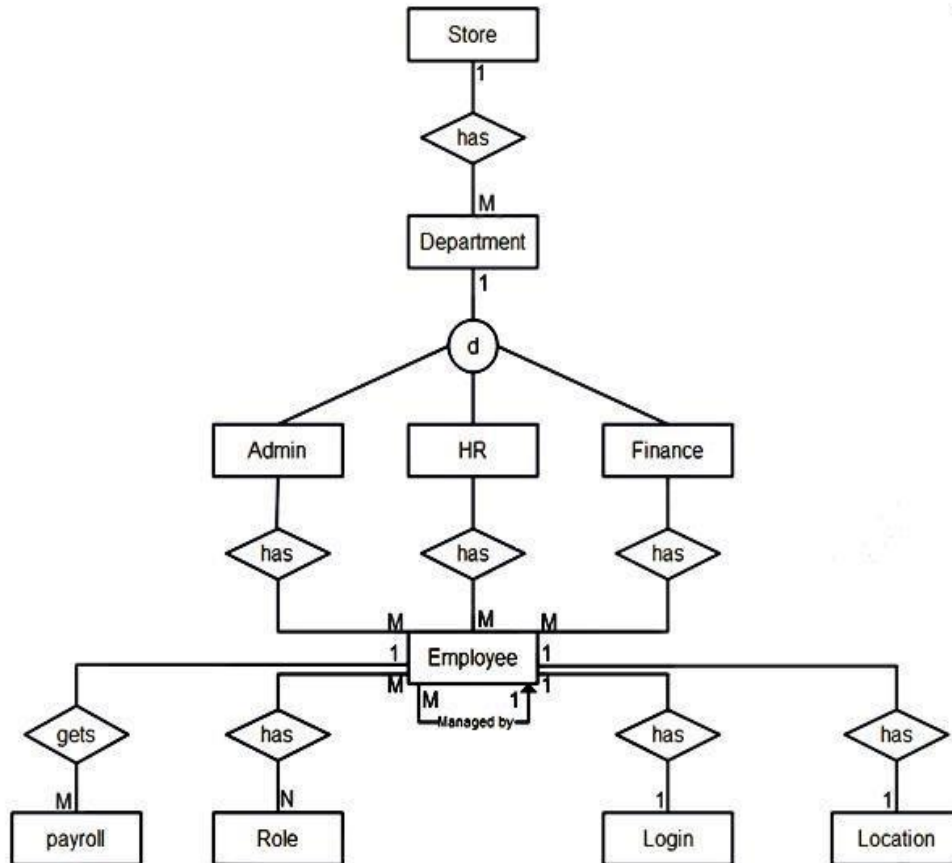
- An Employee may undergo an appraisal with a unique appraisal id.
- Department allocates multiple roles to different employees.
- Department manages multiple leaves and each leave have a unique leave id.

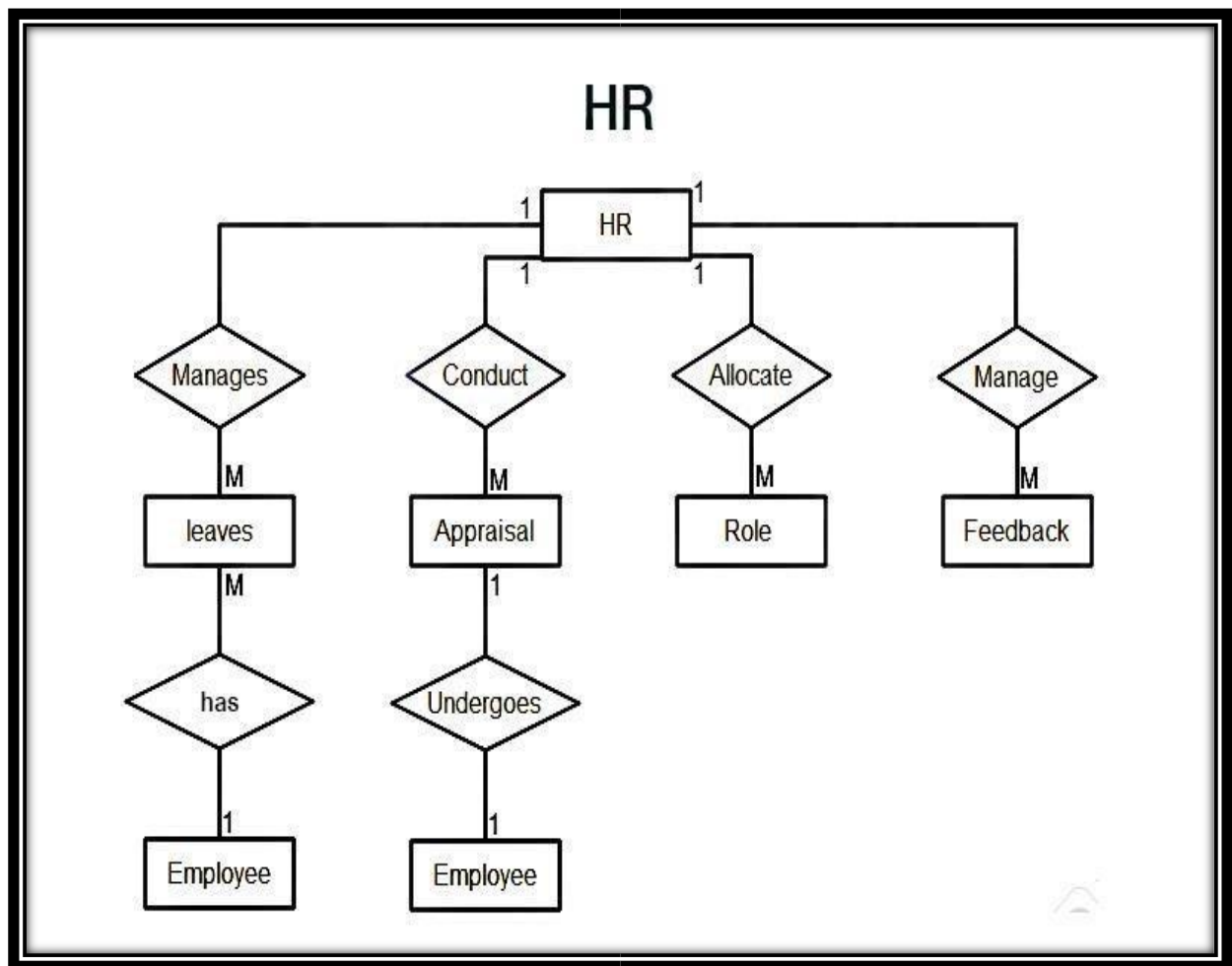
FINANCE:

- Each report having its own unique report id may be generated by multiple invoices, each invoice will be included in only one report.
- Each report will be generated with multiple income statements and each income statement will be included in any one 1 report only.
- Each transaction having unique transaction id will settle multiple payments and each payment will have a particular transaction.
- Payment can be either made through cash or credit card each payment will have a unique payment id.
- Finance department manages transaction having a unique transaction id.
- Multiple transactions create report with a unique report id and 1 transaction can be included in multiple reports.
- One invoice may have multiple products included and one product may be included in multiple invoices.

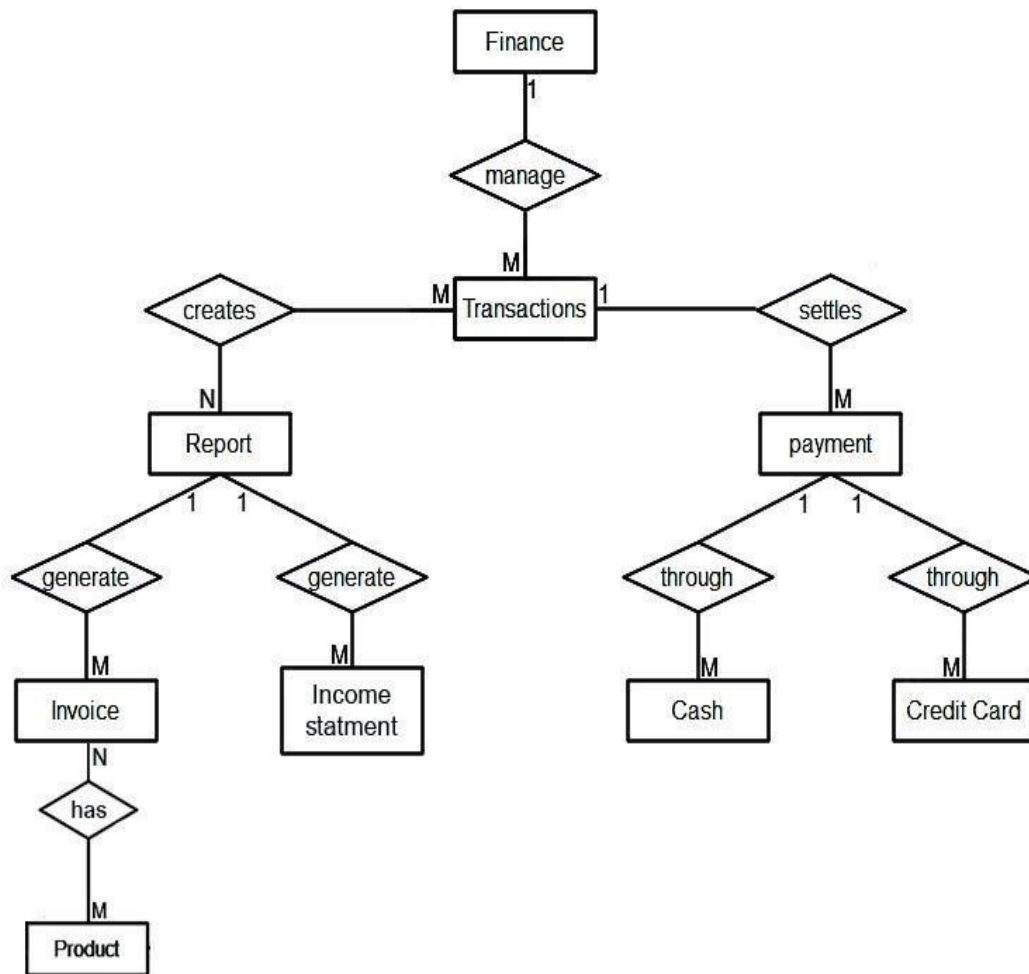
3) Entity Relationship Diagram :

MART MANAGEMENT SYSTEM

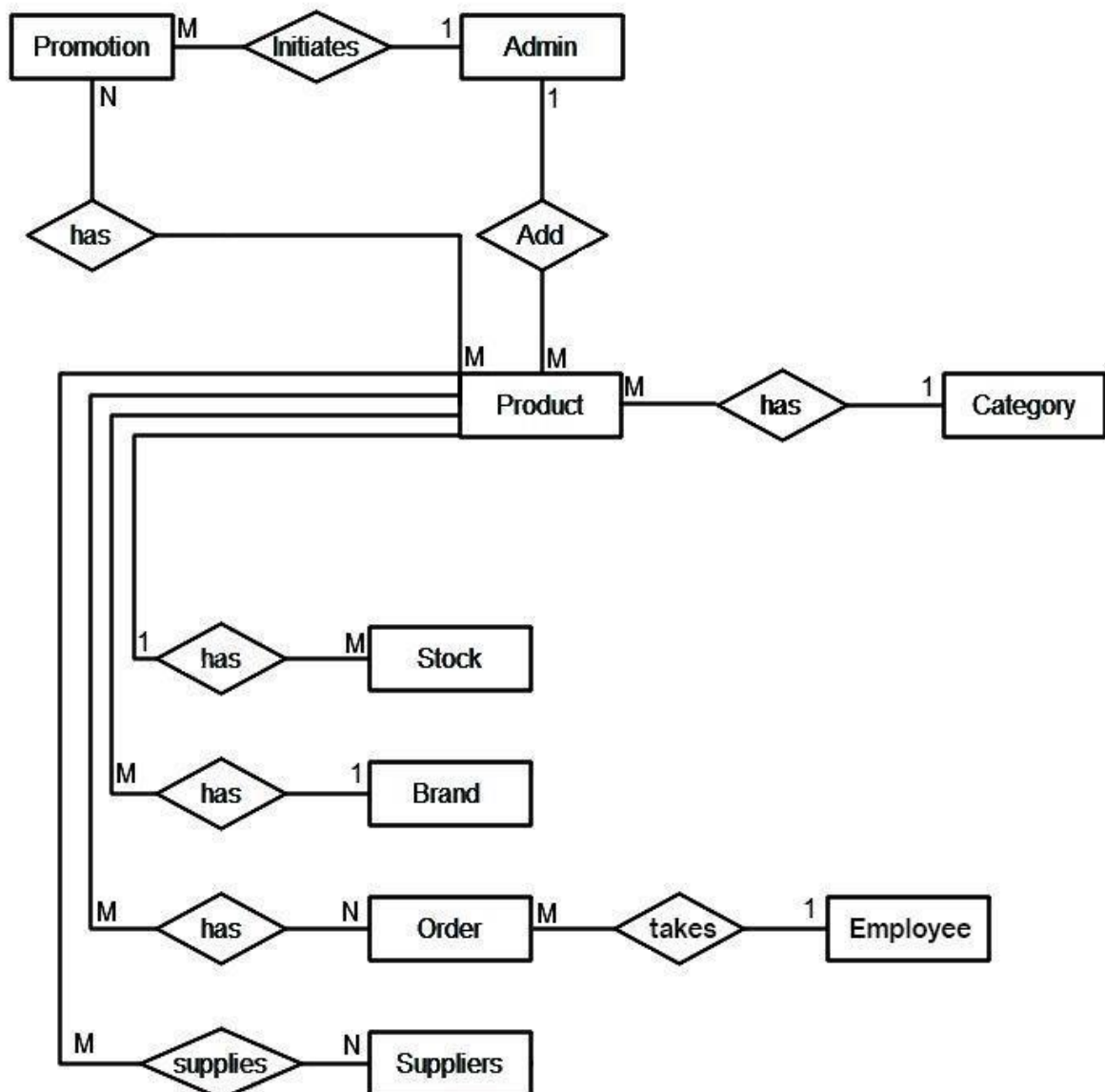




FINANCE

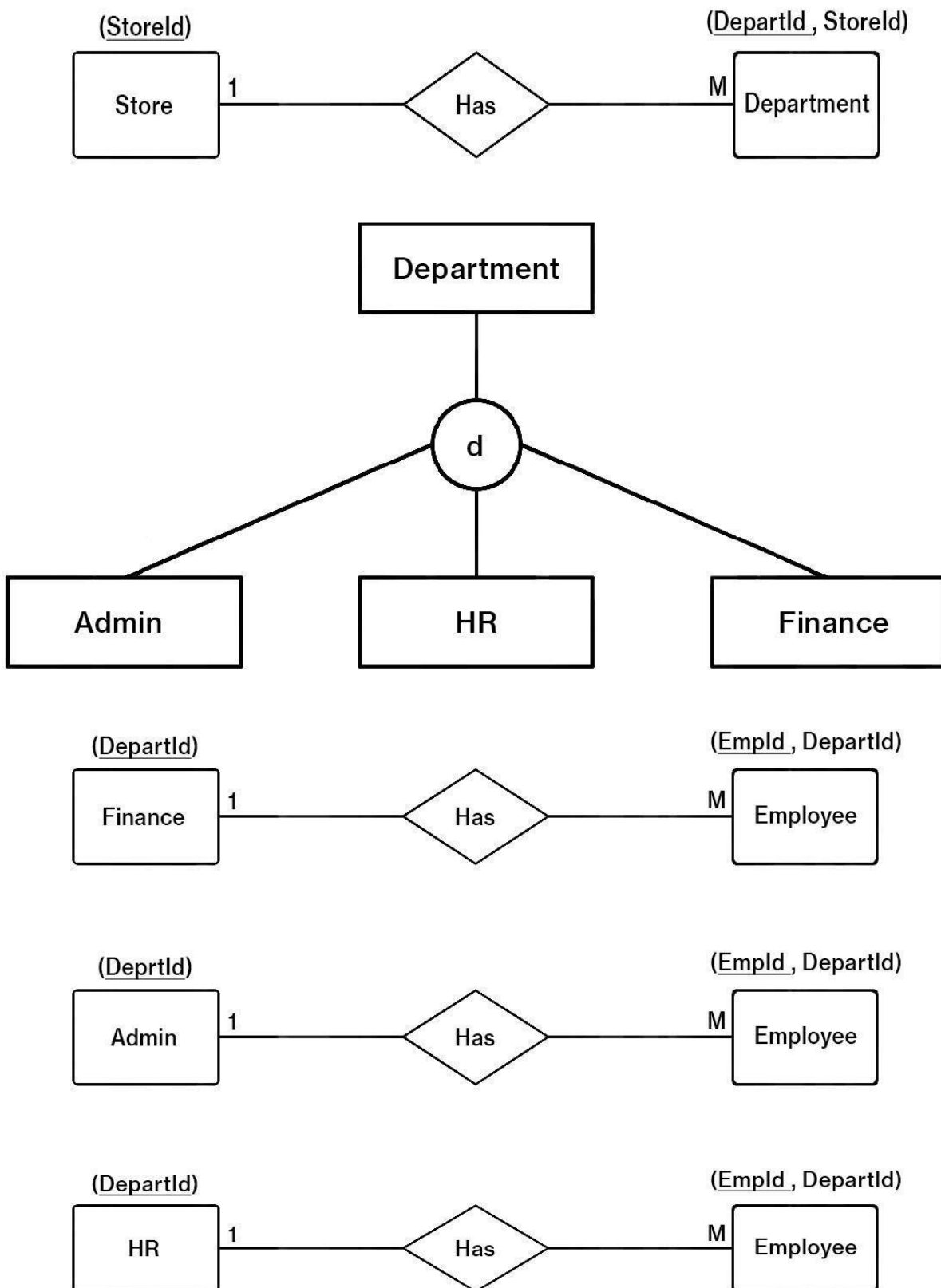


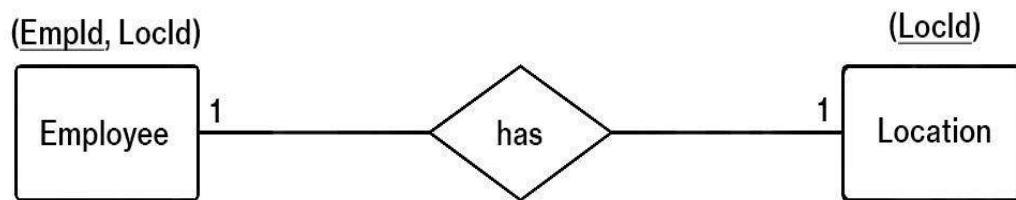
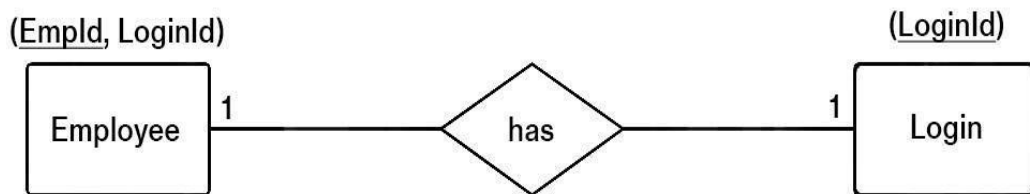
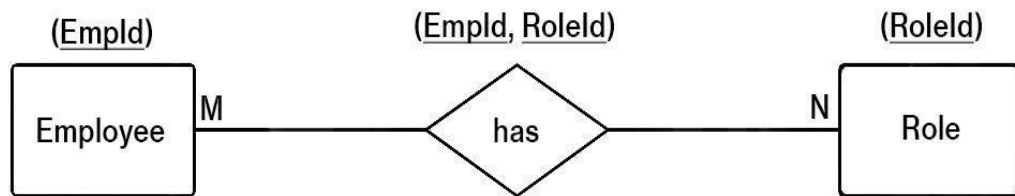
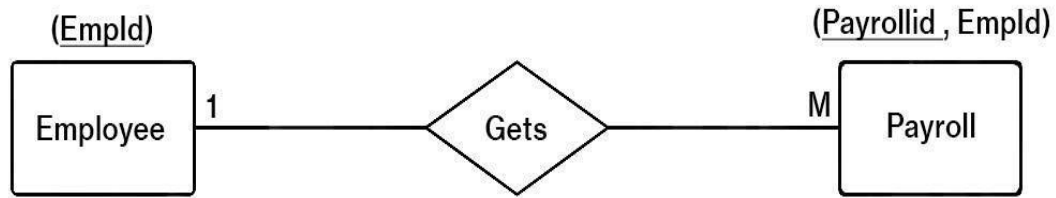
ADMIN



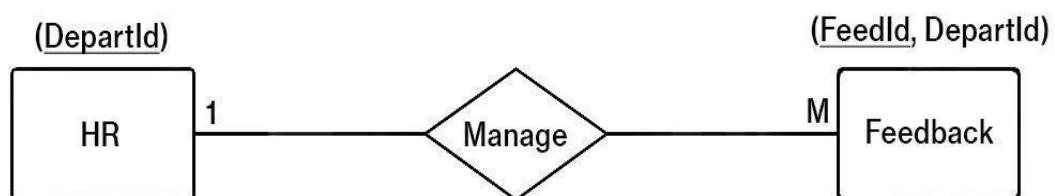
4) Conceptual to Logical Mapping:

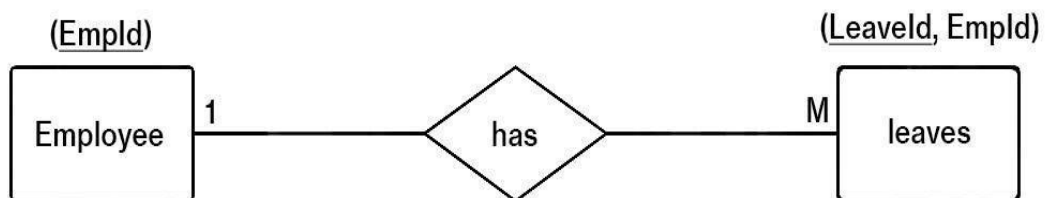
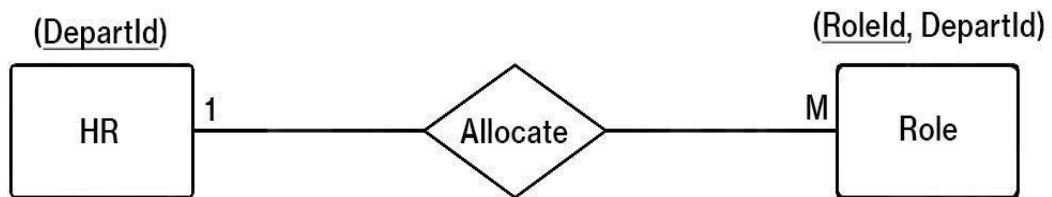
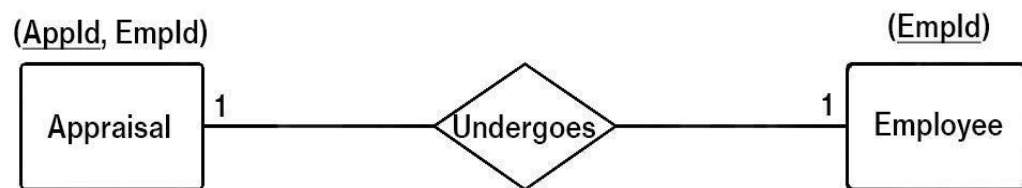
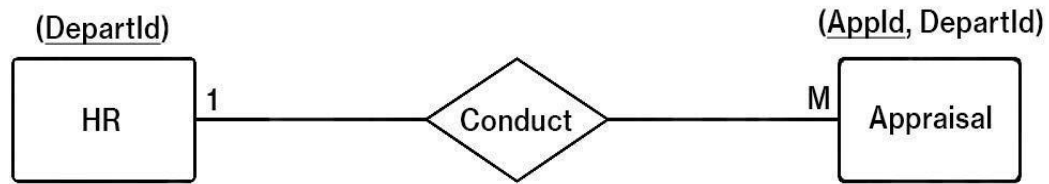
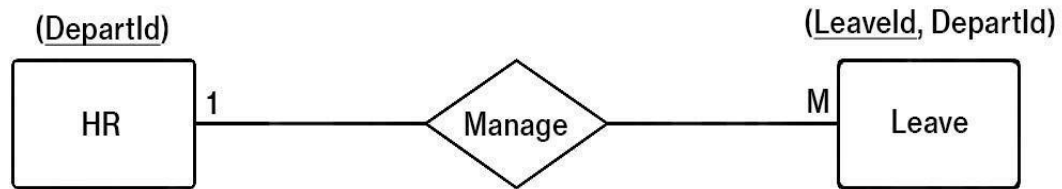
STORE:



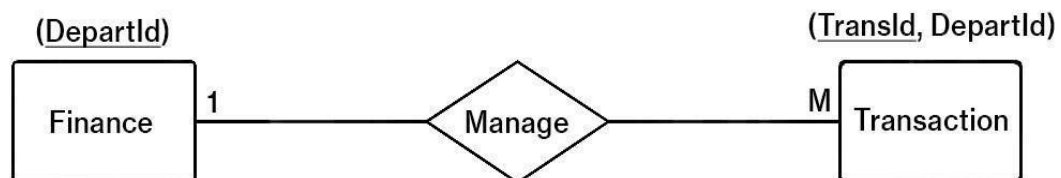
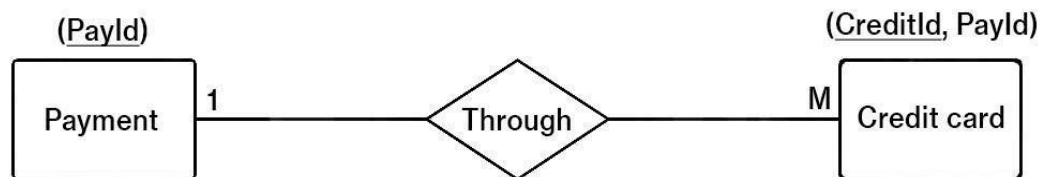
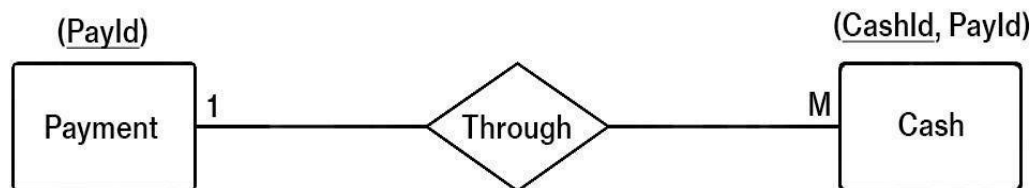
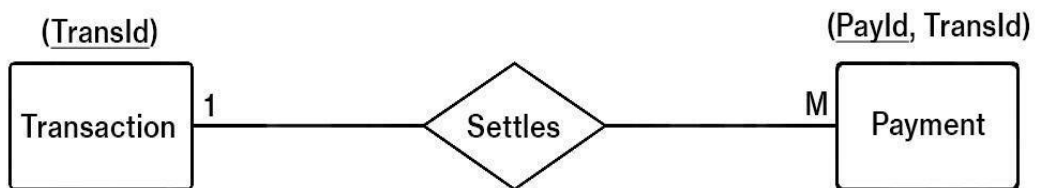
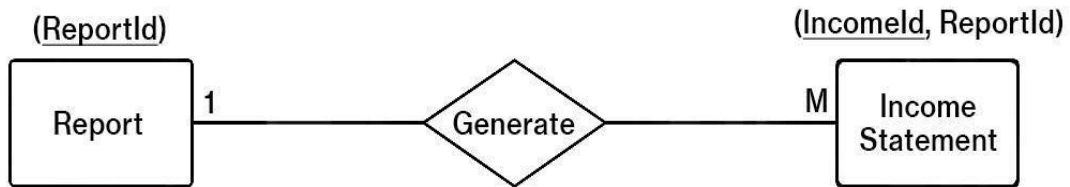
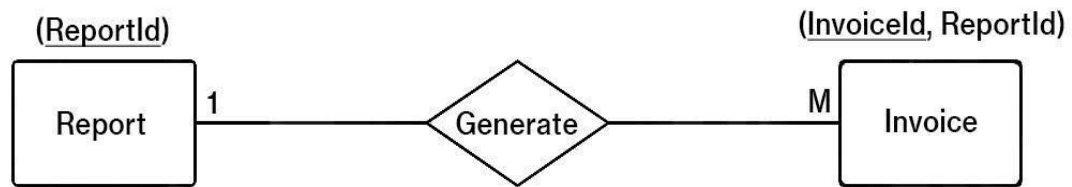


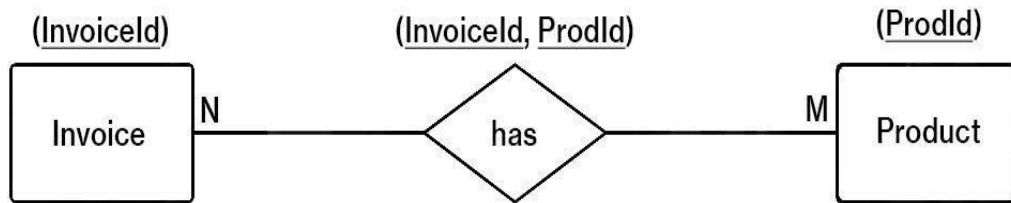
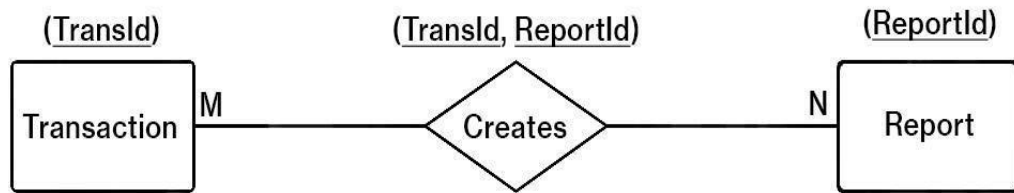
HR:



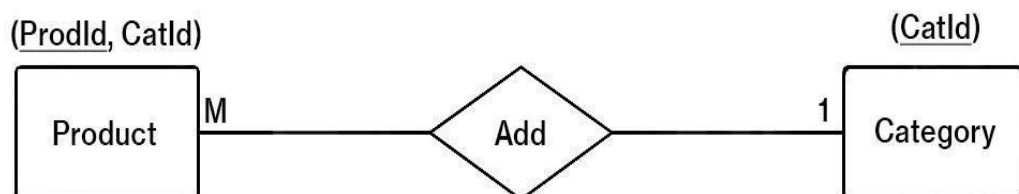
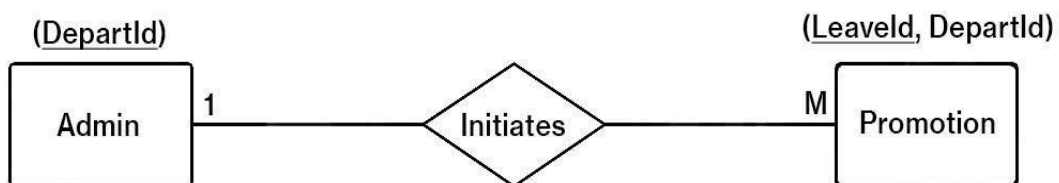
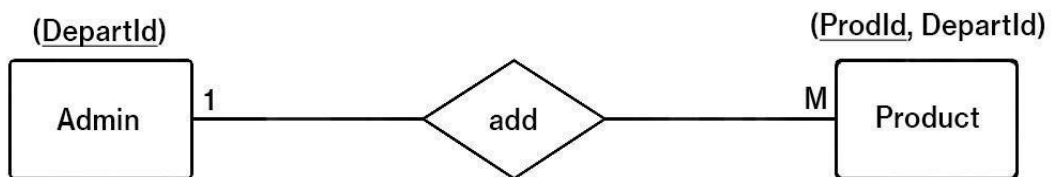


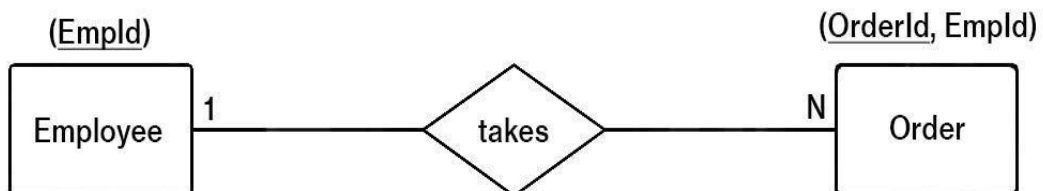
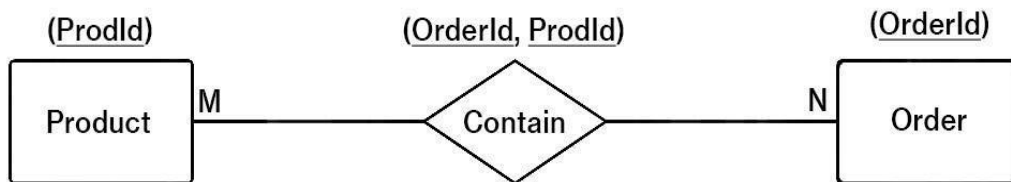
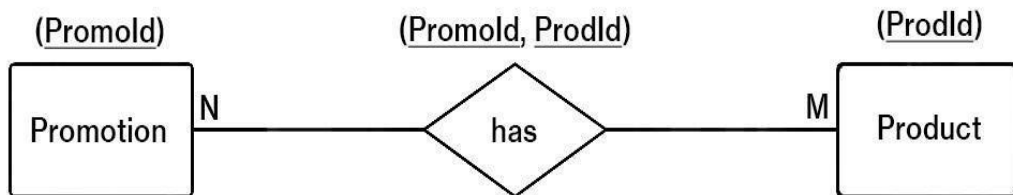
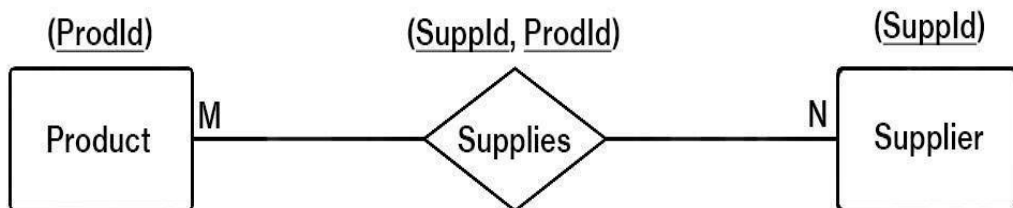
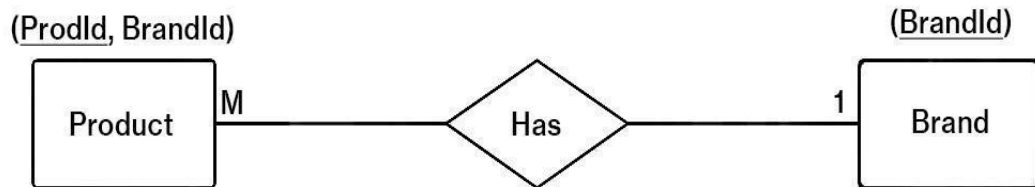
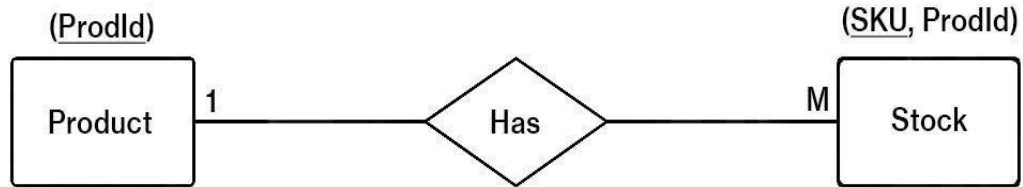
FINANCE:





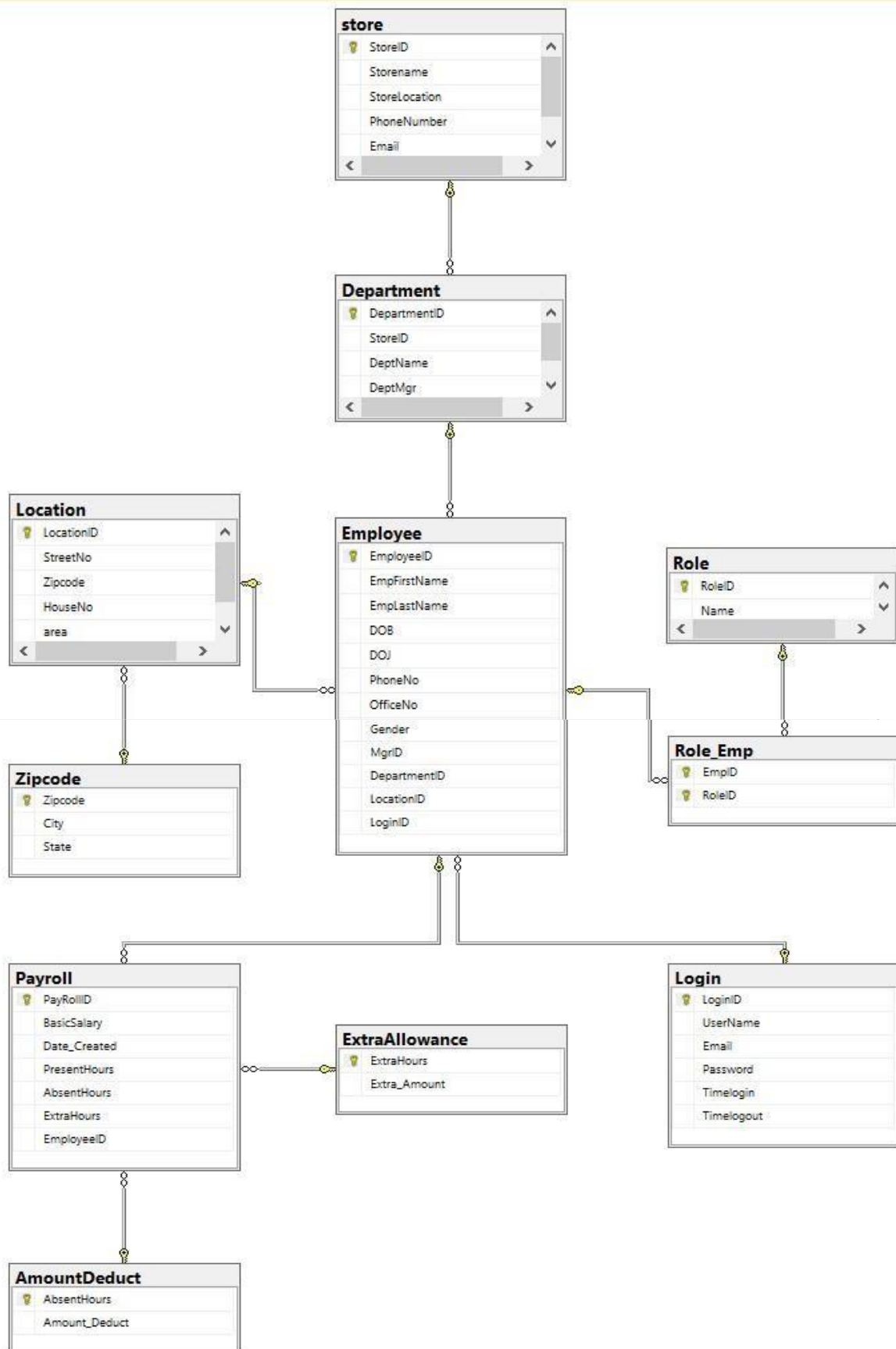
ADMIN:



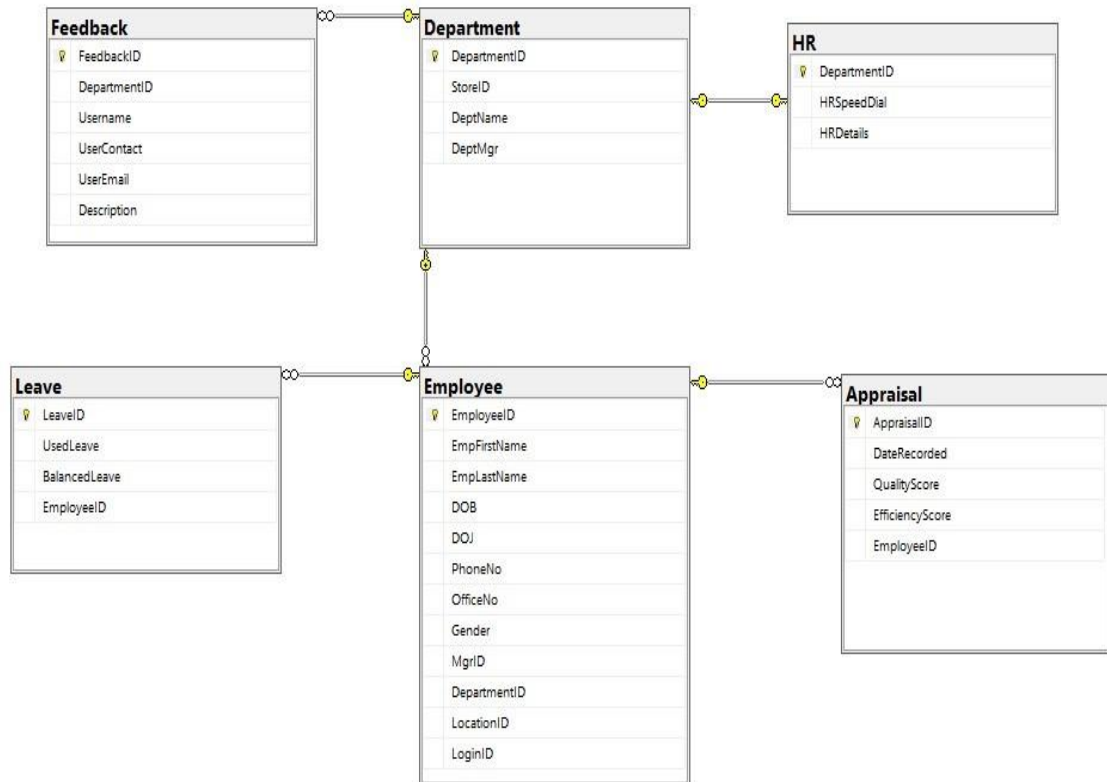


5) Normalized Tables up to BCNF (SQL Server Schema Diagram)

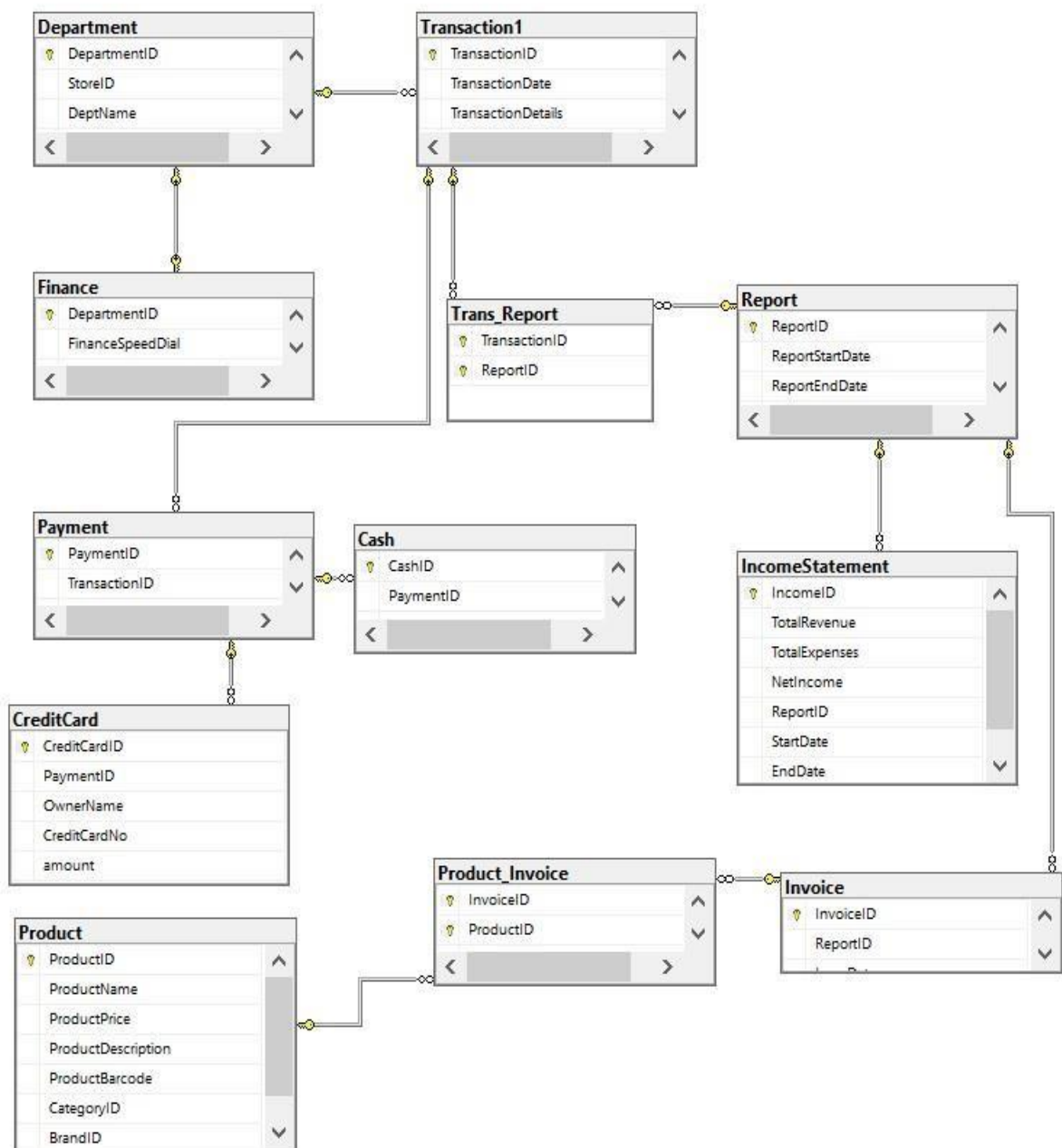
STORE:



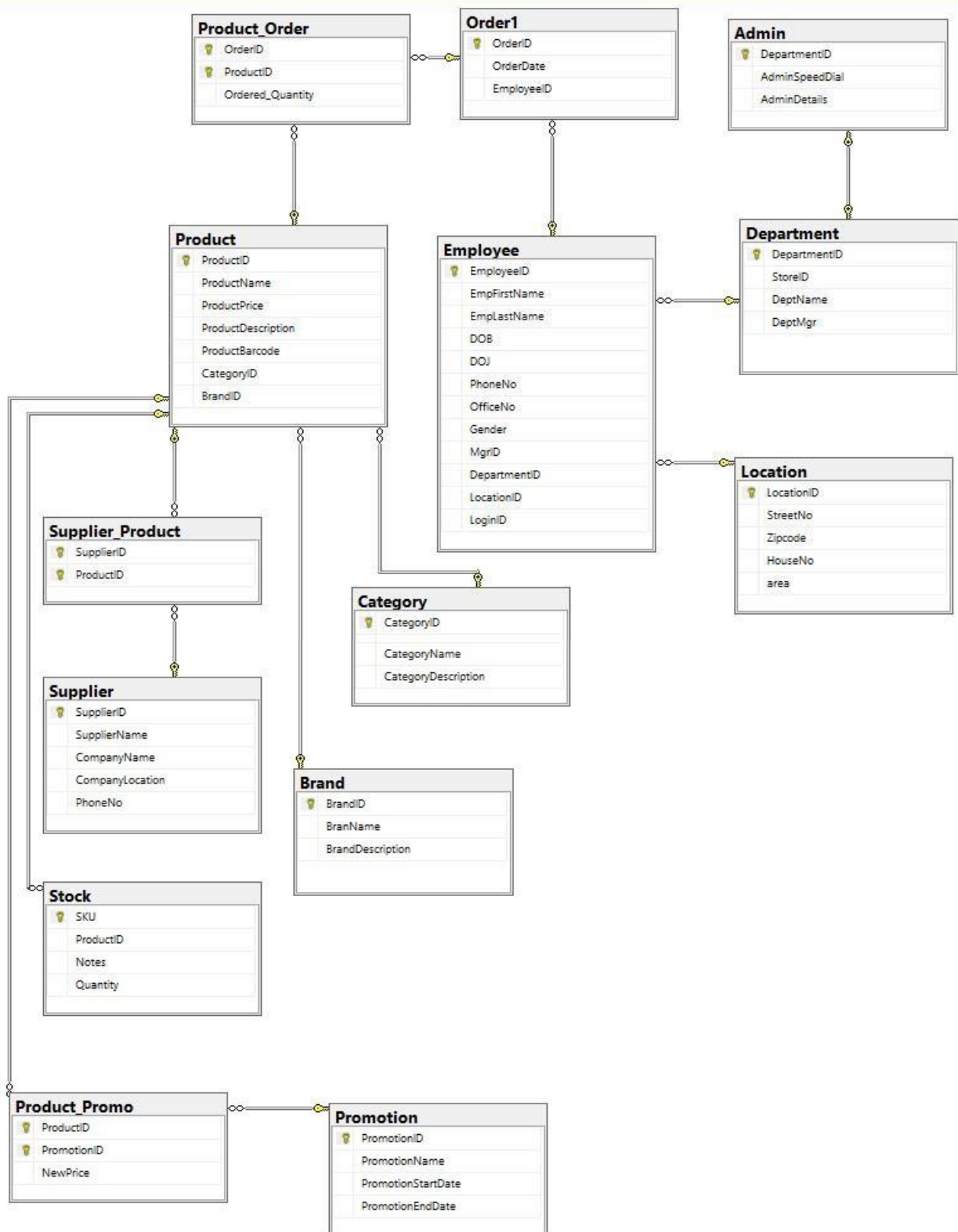
HR:



FINANCE:

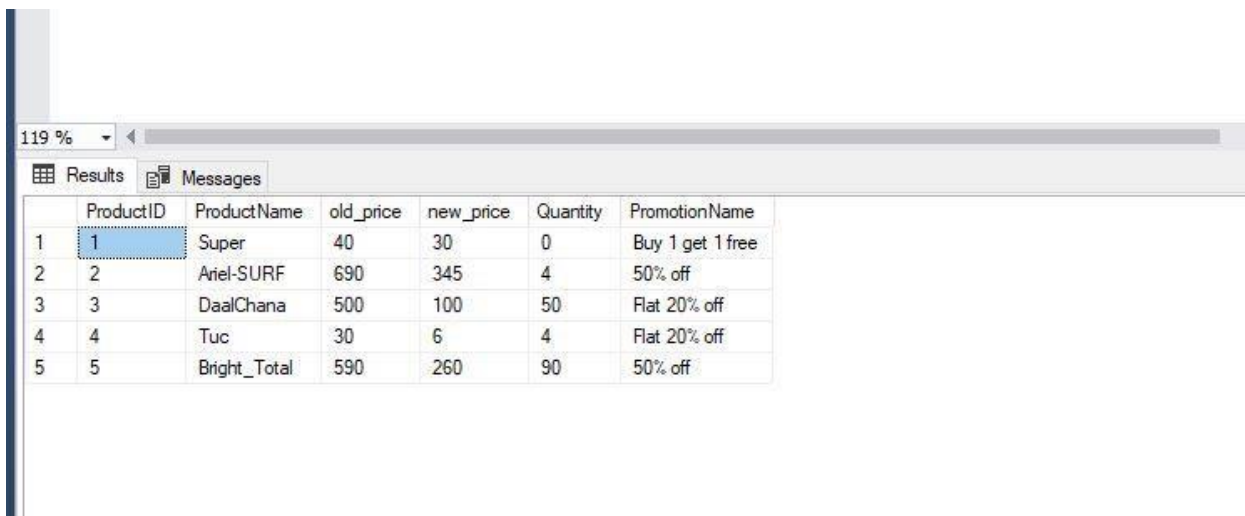


ADMIN:



5)Sample Reports (6 samples) :

- 1) select p.ProductID , p.ProductName ,ProductPrice as old_price , pp.NewPrice as new_price ,
s.Quantity , pr.PromotionName from product p , Product_Promo pp ,
Promotion pr , Category c , Stock s where
p.CategoryID = c.CategoryID and
p.ProductID = s.ProductID and
p.ProductID = pp.ProductID and
pp.PromotionID = pr.PromotionID order
by p.ProductID



	ProductID	ProductName	old_price	new_price	Quantity	PromotionName
1	1	Super	40	30	0	Buy 1 get 1 free
2	2	Ariel-SURF	690	345	4	50% off
3	3	DaalChana	500	100	50	Flat 20% off
4	4	Tuc	30	6	4	Flat 20% off
5	5	Bright_Total	590	260	90	50% off

- 2) select p.ProductID , p.ProductName , su.SupplierName , su.CompanyName
,s.SKU, s.Quantity from product p , Supplier su , Supplier_Product sp ,
Category c , Stock s where
p.CategoryID = c.CategoryID and
p.ProductID = s.ProductID and
p.ProductID = sp.ProductID and
sp.SupplierID = su.SupplierID order
by p.ProductID

119 %						
Results Messages						
	ProductID	ProductName	SupplierName	CompanyName	SKU	Quantity
1	1	Super	Bashir	Bisconi	3	0
2	2	Ariel-SURF	Ahmed	Ariel	2	4
3	3	DaalChana	Bashir	Bisconi	1	50
4	3	DaalChana	Nasir	Ponam	1	50
5	3	DaalChana	Ahmed	Ariel	1	50

3) select e.EmployeeID , e.EmpFirstName + ' ' + e.EmpLastName as Emp_name ,
a.QualityScore ,

a.EfficiencyScore , l.UsedLeave , l.BalancedLeave from

Employee e , Appraisal a , Leave l , Department d

where e.EmployeeID = l.EmployeeID and

d.DepartmentID = e.EmployeeID and

e.EmployeeID = a.EmployeeID

119 %						
Results Messages						
	EmployeeID	Emp_name	QualityScore	EfficiencyScore	UsedLeave	BalancedLeave
1	1	Ashir Azeem	4	5	3	6
2	2	Muhammad Huzaifa	5	4	2	5
3	3	Aqsa Malik	5	2	3	6

4) select e.EmployeeID , e.EmpFirstName + ' ' + e.EmpLastName as emp_name, d.DeptName
, r.Name as

role , p.BasicSalary , l.area , l.Zipcode

from Employee e, Role r, Role_Emp re, Location l , Payroll p , Department d where

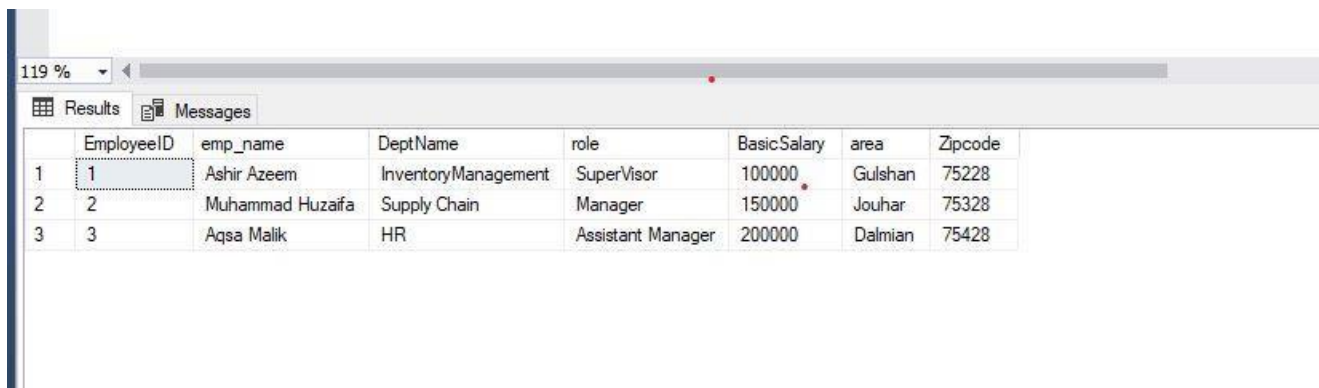
e.EmployeeID = re.EmpID and

re.RoleID = r.RoleID and

e.LocationID = l.LocationID and

e.EmployeeID = p.EmployeeID and

d.DepartmentID = e.DepartmentID



	EmployeeID	emp_name	DeptName	role	BasicSalary	area	Zipcode
1	1	Ashir Azeem	InventoryManagement	SuperVisor	100000	Gulshan	75228
2	2	Muhammad Huzaifa	Supply Chain	Manager	150000	Jouhar	75328
3	3	Aqsa Malik	HR	Assistant Manager	200000	Dalmian	75428

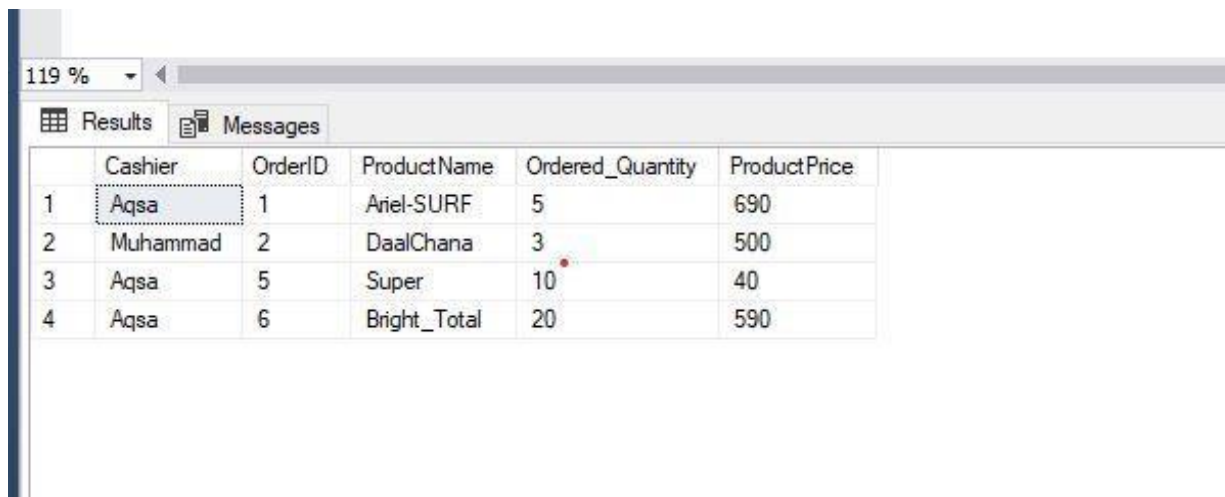
5) select e.EmpFirstName as Cashier , o.OrderID, p.ProductName ,po.Ordered_Quantity ,
p.ProductPrice

from Employee e , Order1 o , Product_Order po , Product p where

e.EmployeeID = o.EmployeeID and

o.OrderID = po.OrderID and

po.ProductID = p.ProductID



	Cashier	OrderID	ProductName	Ordered_Quantity	ProductPrice
1	Aqsa	1	Ariel-SURF	5	690
2	Muhammad	2	DaalChana	3	500
3	Aqsa	5	Super	10	40
4	Aqsa	6	Bright_Total	20	590

6) select e.EmployeeID , e.EmpFirstName +e.EmpLastName as Emp_name ,
p.BasicSalary

,ad.AbsentHours, ad.AbsentHours,ad.Amount_Deduct as ammount_deduct

,ea.ExtraHours,

ea.ExtraHours ,ea.extraAmount as extra_allowance

from Employee e , Role r , Role_Emp re , Payroll p , AmountDeduct ad , ExtraAllowance ea
where

e.EmployeeID = re.EmpID and re.RoleID
 = r.RoleID and
 e.EmployeeID = p.EmployeeID and
 p.AbsentHours = ad.AbsentHours and
 p.ExtraHours = ea.ExtraHours

EmployeeID	Emp_name	BasicSalary	AbsentHours	AbsentHours	ammount_deduct	ExtraHours	ExtraHours	extra_allowance
1	AshirAzeem	100000	1	1	600	5	5	5000
2	MuhammadHuzafa	150000	5	5	3000	3	3	3000
3	AqsaMalik	200000	3	3	1800	4	4	4000

7) Sample DDL/DML/Triggers/Stored Proc/Views/Stored Functions :

DML: (6 samples)

1)

```

----create table Employee(
----EmployeeID int-EmpFirstName varchar(20),
----EmpLastName varchar(20),
----DOB varchar (10),
----DOJ varchar (10),
----PhoneNumber varchar(11),
----OfficeNo varchar(11),
----Gender varchar (4),
----MGRID int,
----DepartmentID int,
----locationid int,
----Loginid varchar(20),
----constraint pk_employee primary key (employeeID),
----constraint fk_employee foreign key (departmentID) references
department(departmentID),
----constraint fk_emp_loc foreign key (locationID) references location(locationID),
----constraint fk_emlogin foreign key (loginid) references login(loginid),
----)
  
```

2)

```

----create table store
  
```

```

----(
----StoreID int,
----Storename varchar(20),
----Storelocation varchar(20),
----PhoneNumber varchar(11),
----Email varchar(20),
----constraint pk_store primary key (StoreID)
----)
3)
----create table department
----(
----DepartmentID int,
----StoreID int,
----DeptName varchar(20),
----DeptMgr varchar(20),
----constraint pk_department primary key (Departmentid),
----constraint fk_store foreign key (StoreID) references store(StoreID)
----)
4)
----create table location
----(
----locationid int,
----StreetNO varchar(10),
----Zipcode varchar(5),
----HouseNo varchar(30),
----area varchar (20),
----constraint pk_location primary key (locationid),
---- constraint fk_zip_code foreign key (zipcode) references zipcode (zipcode)
----)
5)
----- create table Order1(
----- OrderID int,
----- OrderDate varchar(20),
----- Employeeid int,
----- constraint pk_ORDER1 primary key (orderid),
----- constraint fk_order1 foreign key (employeeID) references Employee(employeeID)
----- )

```

6)

```
----- create table ADMIN (
----- DepartmentID int,
----- AdminSpeedDial varchar (11),
----- AdminDetails varchar (30),
----- constraint pk_Admin primary key (DepartmentID),
----- constraint fk_admin foreign key (departmentid) references department (DepartmentID)
----- )
```

DDL: (6 samples)

1)

```
----- Employee -----

select * from Employee
--insert into
--Employee(EmployeeID,DepartmentID,EmpFirstName,EmpLastName,DOB,DOJ,Gender,LocationID,LoginID,MgrID,OfficeNo,PhoneNumber)
--Values
--(1,1,'Ashir','Azeem','24-2-2003','1-2-2021','Male',1,1,null,'123456','03040426130'),
--(2,2,'Muhammad','Huzafa','24-3-2002','1-3-2021','Male',2,2,null,'7891011','03040420015'),
--(3,3,'Aqsa','Malik','24-4-2001','1-4-2021','Female',3,3,null,'121314','03040485472')
```

EmployeeID	EmpFirstName	EmpLastName	DOB	DOJ	PhoneNo	OfficeNo	Gender	MgrID	DepartmentID	LocationID	Logrid
1	Ashir	Azeem	24-2-2003	1-2-2021	03040426130	123456	Male	NULL	1	1	1
2	Muhammad	Huzafa	24-3-2002	1-3-2021	03040420015	7891011	Male	NULL	2	2	2
3	Aqsa	Malik	24-4-2001	1-4-2021	03040485472	121314	Female	NULL	3	3	3

2)

```
----- Department -----

select * from department
--insert into department(DepartmentID,StoreID,DeptName,DeptMgr)
--values
--(2,4,'Supply Chain',3),
--(3,4,'Supply Chain',2)
```

DepartmentID	StoreID	DeptName	DeptMgr
1	2	Inventory Management	5
2	4	Supply Chain	3
3	4	Supply Chain	2

3)

ASHIR-AZEEM-JET.M...ystem - dbo.store project.sql - ASHIR...T\Ashir Azeem (52))

```

INSERT INTO store (StoreID, StoreName, Storelocation, PhoneNumber, Email)
VALUES
(1, '1 General Store', 'Memon Goth', '03334456877', 'generalstore@gmail.com'),
(2, '2 General Store', 'Machhar Colony', '03215869756', 'store@gmail.com'),
(3, '3 general store', 'Nagin Chourangi', '03334154587', 'aqsastore@gmail.com'),
(4, '4 general store', 'Gulistan e jouhar', '033341584587', 'ashirstore@gmail.com'),
(5, '5 general store', 'Malir Colony', '03334156987', 'huzaifastore@gmail.com')

```

157 %

Results Messages

	StoreID	Storename	Storelocation	PhoneNumber	Email
1	1	1 General Store	Memon Goth	03334456877	generalstore@gmail.com
2	2	2 General Store	Machhar Colony	03215869756	store@gmail.com
3	3	3 general store	Nagin Chourangi	03334154587	aqsastore@gmail.com
4	4	4 general store	Gulistan e jouhar	033341584587	ashirstore@gmail.com
5	5	5 general store	Malir Colony	03334156987	huzaifastore@gmail.com

4)

```

--===== login ID =====
select * from login

--insert into login(loginid,UserName,Email>Password,Timelogin,Timeloginout)
--values
--(2, 'huzafa', 'huz@gmail.com', '654321', '10:00 AM', '12:00 AM'),
--(3, 'Aqsa', 'aqsa@gmail.com', '897654', '9:00 AM', '9:00 PM')

```

157 %

Results Messages

	loginid	UserName	Email	Password	Timelogin	Timeloginout
1	1	Ashir Azeem	ashir@gmail.com	123456	11:00 AM	11:00 PM
2	2	huzafa	huz@gmail.com	654321	10:00 AM	12:00 AM
3	3	Aqsa	aqsa@gmail.com	897654	9:00 AM	9:00 PM

5)

```

===== promotion =====
insert into Promotion(PromotionID,promotionName,promotionStartDate,promotionEndDate)
values(1,'Buy 1 get 1 free','11-1-2023','20-1-2023'),
(2,'50% off','20-2-2023','30-2-2023'),
(3,'Flat 20% off','30-3-2023','5-4-2023')
select * from Promotion

```

120 %

Results Messages

	PromotionID	promotionName	promotionStartDate	promotionEndDate
1	1	Buy 1 get 1 free	11-1-2023	20-1-2023
2	2	50% off	20-2-2023	30-2-2023
3	3	Flat 20% off	30-3-2023	5-4-2023

6)

```

===== STOCK =====
insert into stock(SKU,productid,NOTES,quantity)
values(1,3,'Avaialbe','50'),
(2,2,'Short','4'),
(3,1,'NotAvailable','0'),
(4,5,'Avaialbe','90'),
(5,4,'Short','4'),
(6,6,'Available','70')
select * from stock

```

120 %

Results Messages

	SKU	productid	NOTES	quantity
1	1	3	Avaialbe	50
2	2	2	Short	4
3	3	1	NotAvailable	0
4	4	5	Avaialbe	90
5	5	4	Short	4
6	6	6	Available	70

TRIGGERS:

1)

```
----- TRIGGERS -----
create table AdminAudit(
    Details ntext
)
create trigger tr_tblAdmin_forInsert
on Admin
for insert
as
begin
    declare @ID int
    select @ID=DepartmentID from inserted
    insert into AdminAudit
    values ('New DEpartment with id =' +
        cast(@id as nvarchar(5)) +
        'is added at ' + cast(GETDATE() as nvarchar(20))
    )
end
SELECT * FROM AdminAudit
```

109 %

Results Messages

Details	
1	New DEpartment with id =4is added at Jan 15 2023...

2)

```
SQLQuery1.sql - I...Q3\M.HUZAIFA (52))* ✕
create trigger tr_tblAdmin_fordelete
on Admin
for delete
as
begin
    declare @ID int
    select @ID=DepartmentID from deleted
    insert into AdminAudit
    values ('An existing Department with id =' +
        cast(@id as nvarchar(5)) +
        ' is deleted at ' + cast(GETDATE() as nvarchar(20))
    )
end
create trigger tr_tblAdmin_forupdate
on Admin
for update
as
begin
    select * from inserted
    select * from deleted
end
```


3)

```
-----Supplier
create table SupplierAudit(
    Details ntext
)
create trigger tr_tblSup_forInsert
on Supplier
for insert
as
begin
    declare @ID int
    select @ID=SupplierID from inserted
insert into SupplierAudit
values ('New Supplier with id =' + cast(@id as nvarchar(5)) +
    'is added at ' + cast(GETDATE() as nvarchar(20))
)
end
```

4)

```
create trigger tr_tblSup_fordelete
on Supplier
for delete
as
begin
    declare @ID int
    select @ID=SupplierID from deleted
insert into SupplierAudit
values ('An existing Supplier with id =' +
    cast(@id as nvarchar(5)) +
    ' is deleted at ' + cast(GETDATE() as nvarchar(20))
)|
End
```


5)

```
End  
--  
create trigger tr_tblSup_forupdate  
on Supplier  
for update  
as  
begin  
select * from inserted  
select * from deleted  
end
```

6)

```
-----Report  
create table ReportAudit(  
Details ntext  
)  
create trigger tr_tblReport_forInsert  
on Report  
for insert  
as  
begin  
declare @ID int  
select @ID=ReportID from inserted  
insert into ReportAudit  
values ('New Report with id =' +  
cast(@id as nvarchar(5)) +  
'is added at ' + cast(GETDATE() as nvarchar(20))  
)  
end
```

7)

```
-- create trigger tr_tblReport_fordelate
on Report
for delete
as
begin
declare @ID int
select @ID=ReportID from deleted
insert into ReportAudit
values ('An existing Report with id =' +
cast(@id as nvarchar(5)) +
' is deleted at ' + cast(GETDATE() as nvarchar(20))
)
End
```

STORED PROCEDURES:

1)

```
-- create Procedure Emp_App
as
begin
Select a.AppraisalID, e.EmpFirstName,e.EmpLastName,a.QualityScore,a.EfficiencyScore
from employee e, Appraisal a
where e.employeeID=a.employeeID
end
exec Emp_App
```

120 %

Results Messages

	AppraisalID	EmpFirstName	EmpLastName	QualityScore	EfficiencyScore
1	1	Ashir	Azeem	4	5
2	2	Muhammad	Huzaiifa	5	4
3	3	Aqsa	Malik	5	2

2)

```
-- create Procedure Invoice_Prod
-- as
-- begin
-- Select p.ProductName,i.InvoiceID,i.IssueDate
-- from Invoice i, Product p, Product_Invoice pii
-- where pii.InvoiceID=i.InvoiceID and pii.ProductID=p.ProductID
-- end
exec Invoice_Prod
```

120 %

Results Messages

	ProductName	InvoiceID	IssueDate
1	Bright_Total	1	2023-01-15 20:10:15.480
2	Ariel-SURF	2	2023-01-15 20:10:15.480
3	DaalChana	3	2023-01-15 20:10:15.480

FUNCTIONS:

1)

```
create Function Stock_Prod_Cat_Brand()  
returns table  
as  
return(Select  
p.ProductID,p.ProductName,s.SKU,s.Quantity,c.CategoryID,c.CategoryName,b.BrandID,b.BrandName  
from Product p, Stock s, Category c, Brand b  
where p.ProductID=s.ProductID and p.CategoryID=c.CategoryID and b.BrandID=p.BrandID  
)  
Select * from Stock_Prod_Cat_Brand()
```

120 %

Results Messages

	ProductID	ProductName	SKU	Quantity	CategoryID	CategoryName	BrandID	BrandName
1	3	DaalChana	1	50	3	pulses	3	Ponam
2	2	Ariel-SURF	2	4	1	DETERGENT	1	Ariel
3	1	Super	3	0	2	COOKIES	2	Bisconi
4	5	Bright_Total	4	90	1	DETERGENT	1	Ariel
5	4	Tuc	5	4	2	COOKIES	2	Bisconi
6	6	DaalMasoor	6	70	3	pulses	3	Ponam

2)

```
create Function Payment_Method()  
returns table  
as  
return(Select p.PaymentID,cc.CreditCardID,cc.OwnerName,cc.amount  
from Payment p, CreditCard cc  
where p.PaymentID=cc.PaymentID  
)  
Select * from Payment_Method()
```

120 %

Results Messages

	PaymentID	CreditCardID	OwnerName	amount
1	3	1	Hamza	100000000
2	2	2	Bilal	20000000
3	1	3	Imran	30000000

3)

```

create Function T_Sal(@EmpID int)
returns table
as
return(Select e.EmpFirstName,e.EmpLastName,p.BasicSalary,p.ExtraHours,ea.ExtraAmount,
(p.BasicSalary+ea.ExtraAmount) as T_Sal
from Employee e, Payroll p, ExtraAllowance ea
where e.EmployeeID=p.EmployeeID and ea.ExtraHours=p.ExtraHours and
e.EmployeeID=@EmpID
)
Select * from T_Sal(3)

```

120 %

Results Messages

	EmpFirstName	EmpLastName	BasicSalary	ExtraHours	ExtraAmount	T_Sal
1	Aqsa	Malik	200000	4	4000	200004000

VIEWS:

1)

```

===== VIEWS =====
--create view emp_vu
--as
--Select e.*, d.DeptName from Employee e, Department d where e.DepartmentID=d.DepartmentID
--and d.DepartmentID=1
--select * from emp_vu

```

120 %

Results Messages

	EmployeeID	EmpFirstName	EmpLastName	DOB	DOJ	PhoneNumber	OfficeNo	Gender	MGRID	DepartmentID	locationid	Loginid	DeptName
1	1	Ashir	Azeem	24-2-2003	1-2-2021	03040426130	123456	Male	NULL	1	1	1	InventoryManagement

2)

```
--create view product_vu as
--Select p.*, c.CategoryName from Product p, Category c where p.CategoryID=c.CategoryID and
--c.CategoryID=2
--select * from product_vu
```

	ProductID	ProductName	ProductPrice	ProductDescription	ProductBarCode	CategoryID	BrandID	CategoryName
1	1	Super	40	Made from Eggs	13354	2	2	COOKIES
2	4	Tuc	30	Salty_Biscuit	12224	2	2	COOKIES

3)

```
--create view IncomeStatement_vu as
--Select * from Income_Statement
```

```
--Select * from IncomeStatement_vu
```

	IncomeID	Total_Revenue	Total_Expenses	NetIncome	ReportID	StartDate	EndDate
1	1	80000	750000	960000	1	25-6-2021	30-12-2023
2	2	90000	80000	100000	2	14-5-2020	30-5-2023
3	3	70000	750000	860000	3	21-3-2019	30-6-2023