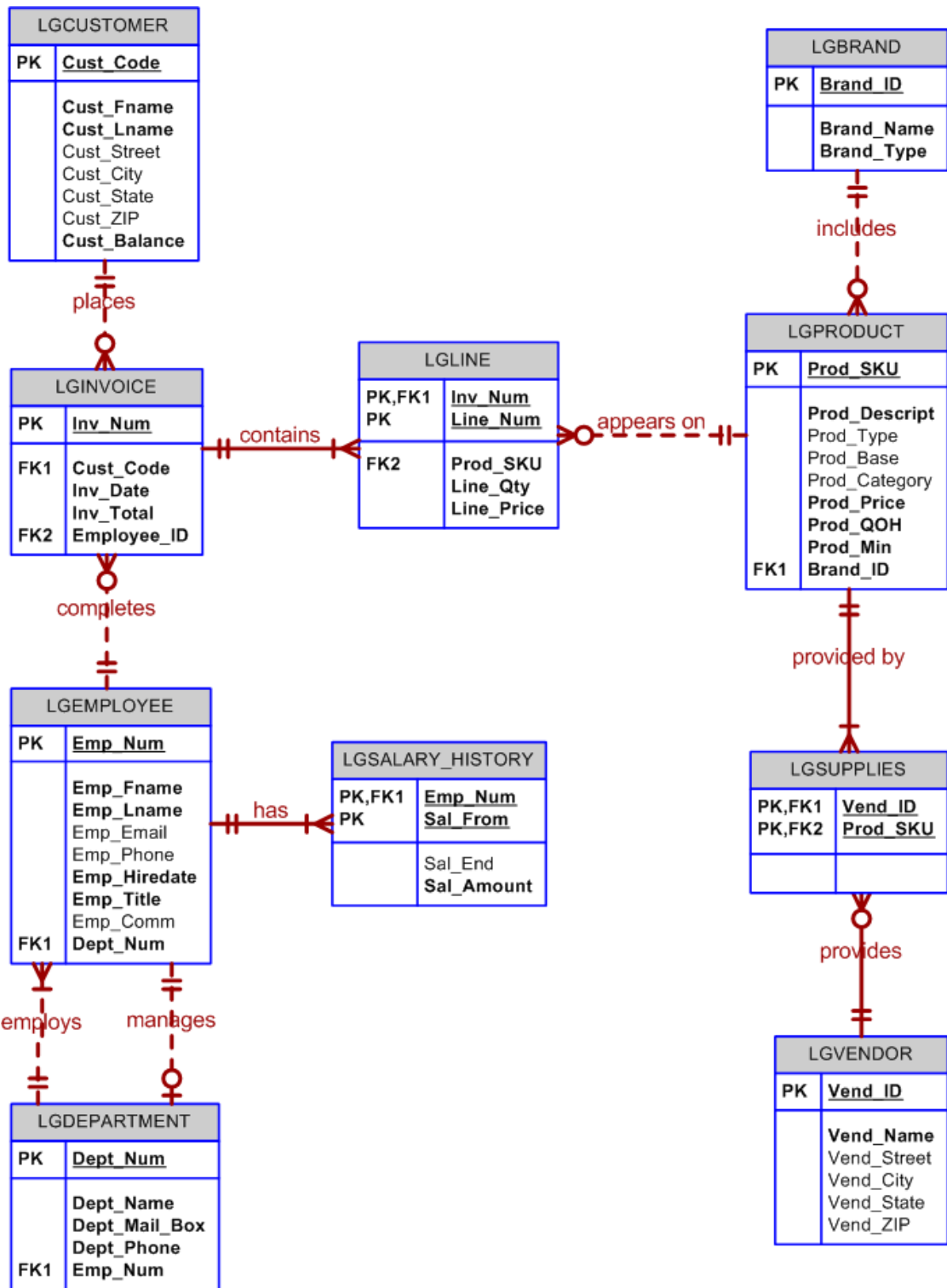


SQL Part 2

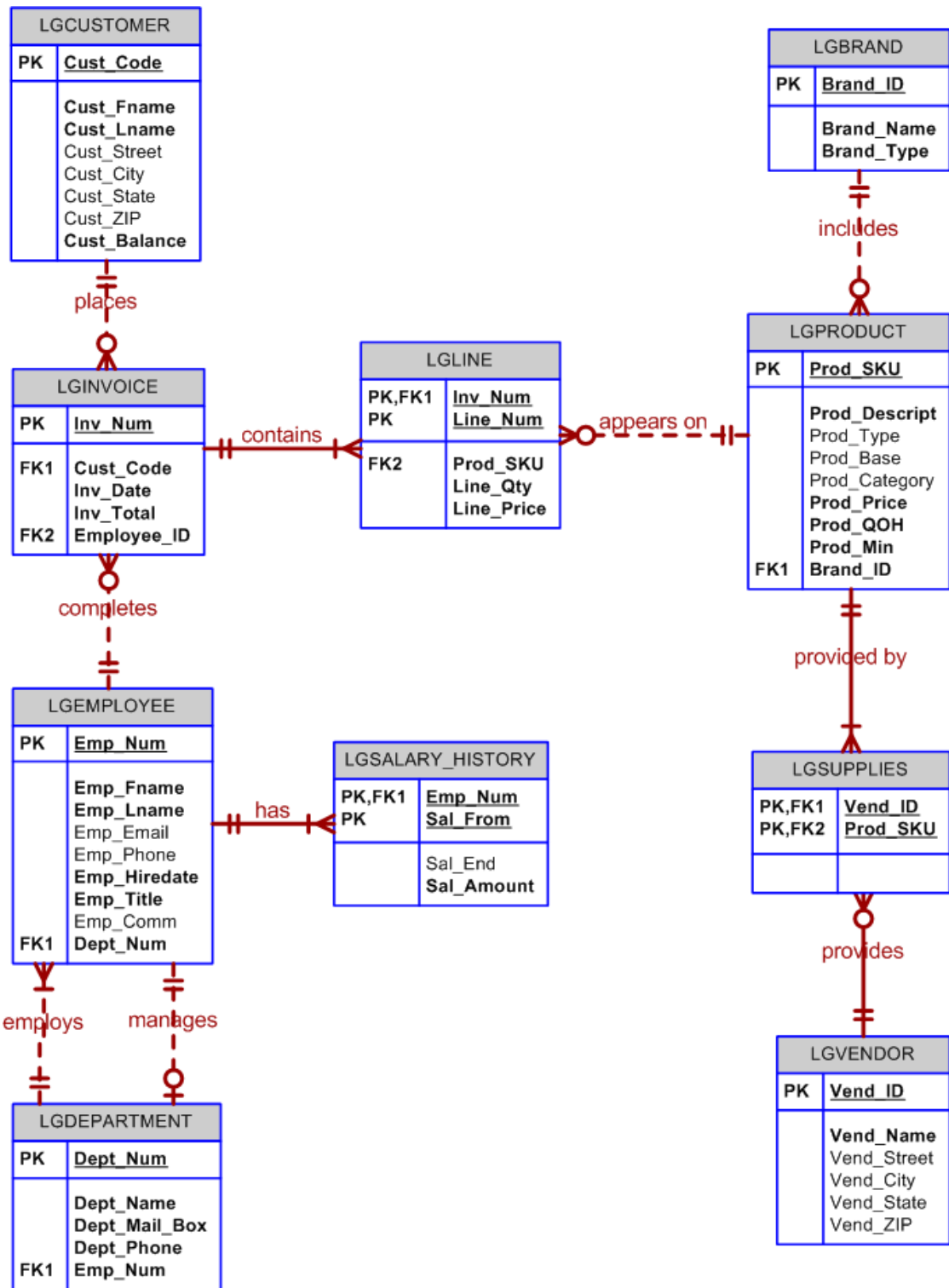
Ch 8 Exercises 20-27 in the Textbook



•Write a query to display the current salary for each employee in department 300. Assume that only current employees are kept in the system, and therefore the most current salary for each employee is the entry in the salary history with a NULL end date. Sort the output in descending order by salary amount. (Figure P8.20)

Figure P8.20 Current salary for employees in department 300

Emp_Num	Emp_LName	Emp_FName	Sal_Amount
83746	RANKIN	SEAN	95550
84328	CARPENTER	FERN	94090
83716	RIVERA	HENRY	85920
84432	JAMISON	MERLE	85360
83902	VARGAS	ROCKY	79540
83695	MENDEZ	CARROLL	79200
84500	WESTON	CHRISTINE	78690
84594	TIDWELL	ODELL	77400
83910	AVERY	LAUREN	76110
83359	WATTS	MERLE	72240
83790	ACEVEDO	LAVINA	72000



•Write a query to display the current salary for each employee in department 300. Assume that only current employees are ke
Figure P8.20 Current salary for employees in department 300

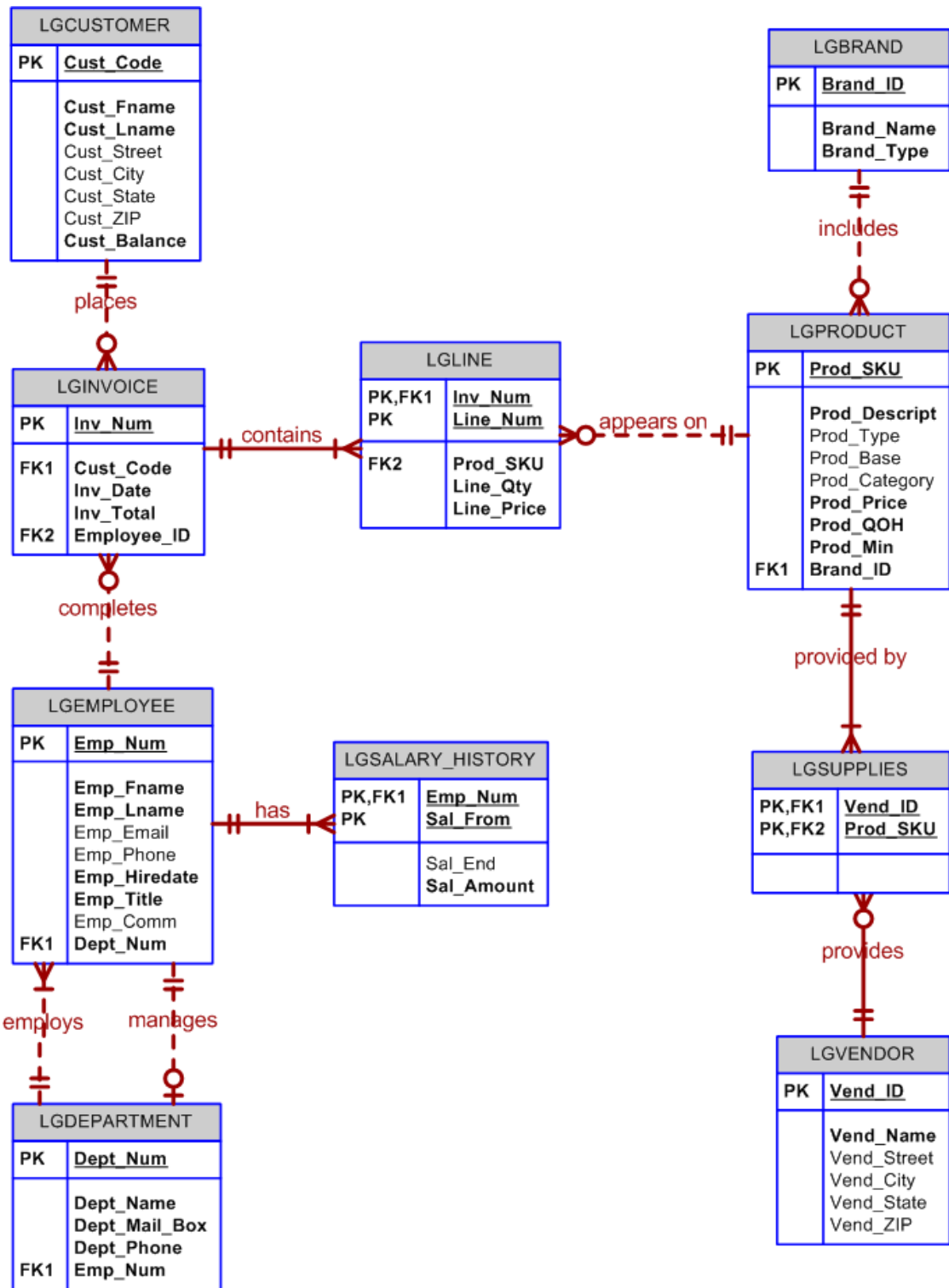
Emp_Num	Emp_LName	Emp_FName	Sal_Amount
83746	RANKIN	SEAN	95550
84328	CARPENTER	FERN	94090
83716	RIVERA	HENRY	85920
84432	JAMISON	MERLE	85360
83902	VARGAS	ROCKY	79540
83695	MENDEZ	CARROLL	79200
84500	WESTON	CHRISTINE	78690
84594	TIDWELL	ODELL	77400
83910	AVERY	LAUREN	76110
83359	WATTS	MERLE	72240
83790	ACEVEDO	LAVINA	72000

```
SELECT e.emp_num, emp_lname, emp_fname,  
       sal_amount  
FROM lgemployee e JOIN lgsalary_history s ON  
e.emp_num = s.emp_num  
WHERE sal_end IS NULL  
AND dept_num = 300  
ORDER BY sal_amount DESC;
```

•Write a query to display the starting salary for each employee. The starting salary would be the entry in the salary history with the oldest salary start date for each employee. Sort the output by employee number. (Figure P8.21)

Figure P8.21 Starting salary for each employee

Emp_Num	Emp_LName	Emp_FName	Sal_Amount
83304	MCDONALD	TAMARA	19770
83308	LOVE	CONNIE	11230
83312	BAKER	ROSALBA	39260
83314	DAVID	CHAROLETTE	15150
83318	PECK	DARCIE	22330
83321	FARMER	ANGELINA	18250
83332	LONG	WILLARD	23380
83341	CORTEZ	CHRISTINE	14510
83347	WINN	QUINTIN	17010
83349	SINGH	JENNIFFER	21220
83359	WATTS	MERLE	25370
83366	BLEDSE	PHOEBE	23200



•write a query to display the starting salary for each employee. The starting salary would be the entry in the salary history with the lowest salary.
Figure P8.21 Starting salary for each employee

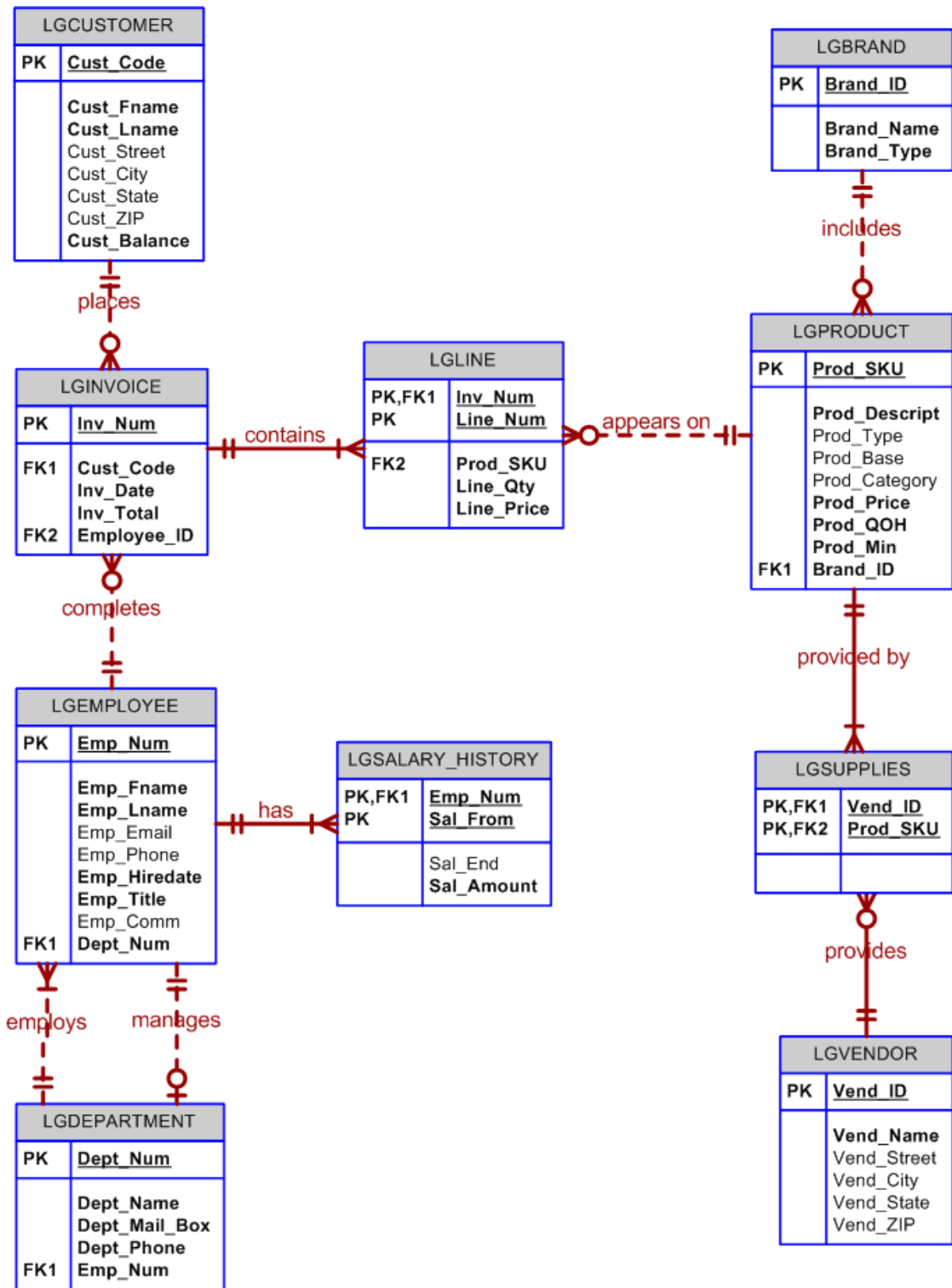
Emp_Num	Emp_LName	Emp_FName	Sal_Amount
83304	MCDONALD	TAMARA	19770
83308	LOVE	CONNIE	11230
83312	BAKER	ROSALBA	39260
83314	DAVID	CHAROLETTE	15150
83318	PECK	DARCIE	22330
83321	FARMER	ANGELINA	18250
83332	LONG	WILLARD	23380
83341	CORTEZ	CHRISTINE	14510
83347	WINN	QUINTIN	17010
83349	SINGH	JENNIFFER	21220
83359	WATTS	MERLE	25370
83366	BLEDSE	PHOEBE	23200

```
SELECT e.emp_num, emp_lname, emp_fname, sal_amount
FROM lmployee e join lgsalary_history s ON e.emp_num = s.emp_num
WHERE sal_from = (SELECT Min(sal_from)
                  FROM lgsalary_history s2 WHERE e.emp_num = s2.emp_num)
ORDER BY e.emp_num;
```


•Write a query to display the invoice number, line numbers, product SKUs, product descriptions, and brand ID for sales of sealer and top coat products of the same brand on the same invoice. (Figure P8.22)

Figure P8.22 Invoices for sealer and top coat of the same brand

Inv_Num	I.Line_Num	p.Prod_Sku	p.Prod_Descript	I2.Line_Num	p2.Prod_Sku	p2.Prod_Descript	Brand_ID
115	2	5140-RTG	Fire Resistant Sealer, for Exterior Wood (ULC Approved)	1	1203-AIS	Fire Retardant Coating, Latex, Interior, Flat (ULC Approved)	35
118	2	5140-RTG	Fire Resistant Sealer, for Exterior Wood (ULC Approved)	5	5046-TTC	Aluminum Paint, Heat Resistant (Up to 427°C - 800°F)	35
135	5	3036-PCT	Sealer, for Knots	2	1074-VVJ	Light Industrial Coating, Exterior, Water Based ('eggshell-like' - MPI Gloss Level 3)	25
153	2	3701-YAW	Sealer, Solvent Based, for Concrete Floors	1	3955-NWD	Water Repellant, Clear (Not Paintable)	30
222	1	1336-FVM	Alkyd, Sanding Sealer, Clear	3	8199-YRF	Varnish, Exterior, Water Based, (Satin-Like) MPI Gloss Level 4	33
234	4	5728-ZPO	Shop Coat, Quick Dry, for Interior Steel	3	9272-LTP	Varnish, Marine Spar, Exterior, Gloss (MPI Gloss Level 6)	27
234	4	5728-ZPO	Shop Coat, Quick Dry, for Interior Steel	2	9126-PWF	Latex, Recycled (Consolidated), Interior (MPI Gloss Level 3)	27
243	1	4072-SWV	Sealer, Solvent Based, for Concrete Floors	3	5653-RTU	Aluminum Paint	23
287	1	8894-LUR	Lacquer, Sanding Sealer, Clear	5	9838-FUF	Fire Retardant Top-Coat, Clear, Alkyd, Interior (ULC Approved)	27
333	1	3701-YAW	Sealer, Solvent Based, for Concrete Floors	6	2584-CIJ	Stain, for Exterior Wood Decks	30
333	1	3701-YAW	Sealer, Solvent Based, for Concrete Floors	5	4784-SLU	Lacquer, Clear, Flat	30
369	2	1403-TUY	Sealer, Water Based, for Concrete Floors	1	8726-ZNM	Floor Paint, Alkyd, Low Gloss	29



Write a query to display the invoice number, line numbers, product SKUs, product descriptions, and brand ID for sales of sealer and top coat of the same brand

Figure P8.22 Invoices for sealer and top coat of the same brand

Inv_Num	I.Line_Num	p.Prod_Sku	p.Prod_Descript	I2.Line_Num	p2.Prod_Sku	p2.Prod_Descript	Brand_ID
115	2	5140-RTG	Fire Resistant Sealer, for Exterior Wood (ULC Approved)	1	1203-AIS	Fire Retardant Coating, Latex, Interior, Flat (ULC Approved)	35
118	2	5140-RTG	Fire Resistant Sealer, for Exterior Wood (ULC Approved)	5	5046-TTC	Aluminum Paint, Heat Resistant (Up to 427°C - 800°F)	35
136	5	3036-PCT	Sealer, for Knots	2	1074-VVJ	Light Industrial Coating, Exterior, Water Based ('eggshell-like' - MPI Gloss Level 3)	25
153	2	3701-YAW	Sealer, Solvent Based, for Concrete Floors	1	3955-NWD	Water Repellant, Clear (Not Paintable)	30
222	1	1336-FVM	Alkyd, Sanding Sealer, Clear	3	8199-YRF	Varnish, Exterior, Water Based, (Satin-Like) MPI Gloss Level 4	33
234	4	5728-ZPO	Shop Coat, Quick Dry, for Interior Steel	3	9272-LTP	Varnish, Marine Spar, Exterior, Gloss (MPI Gloss Level 6)	27
234	4	5728-ZPO	Shop Coat, Quick Dry, for Interior Steel	2	9126-PWF	Latex, Recycled (Consolidated), Interior (MPI Gloss Level 3)	27
243	1	4072-SWV	Sealer, Solvent Based, for Concrete Floors	3	5653-RTU	Aluminum Paint	23
287	1	8894-LUR	Lacquer, Sanding Sealer, Clear	5	9838-FUF	Fire Retardant Top-Coat, Clear, Alkyd, Interior (ULC Approved)	27
333	1	3701-YAW	Sealer, Solvent Based, for Concrete Floors	6	2584-CU	Stain, for Exterior Wood Decks	30
333	1	3701-YAW	Sealer, Solvent Based, for Concrete Floors	5	4784-SLU	Lacquer, Clear, Flat	30
369	2	1403-TUY	Sealer, Water Based, for Concrete Floors	1	8726-ZNM	Floor Paint, Alkyd, Low Gloss	29

```

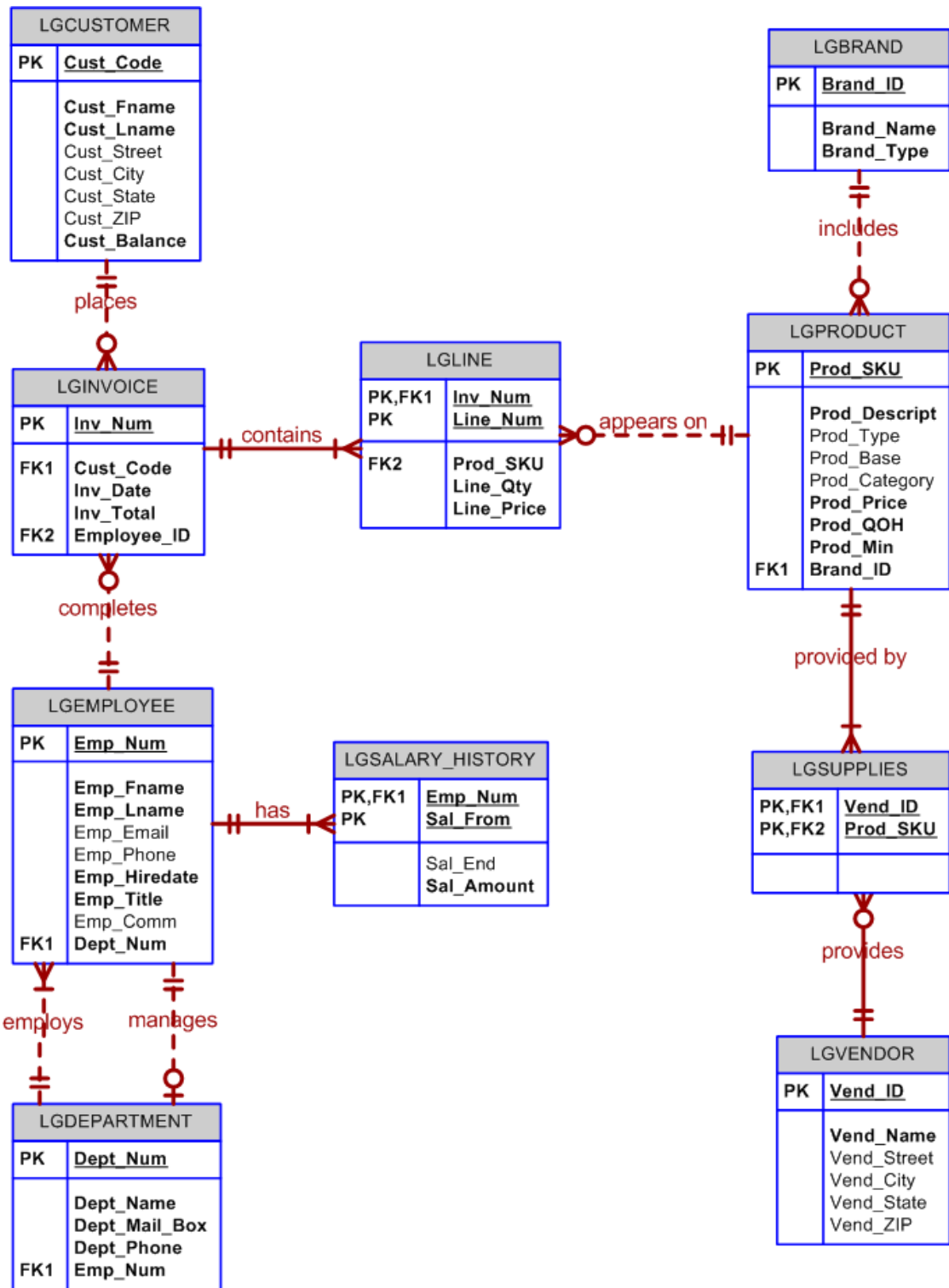
SELECT I.inv_num, I.line_num, p.prod_sku, p.prod_descript, I2.line_num, p2.prod_sku,
p2.prod_descript, p.brand_id
FROM (Igline I join Igproduct p ON I.prod_sku = p.prod_sku) join
      (Igline I2 join Igproduct p2 ON I2.prod_sku = p2.prod_sku)
ON I.inv_num = I2.inv_num
WHERE p.brand_id = p2.brand_id
      AND p.prod_category = 'Sealer'
      AND p2.prod_category = 'Top Coat'
ORDER BY I.inv_num, I.line_num;

```

•The Binder Prime Company wants to recognize the employee who sold the most of their products during a specified period. Write a query to display the employee number, employee first name, employee last name, e-mail address, and total units sold for the employee who sold the most Binder Prime brand products between November 1, 2015, and December 5, 2015. If there is a tie for most units sold, sort the output by employee last name. (Figure P8.23)

Figure P8.23 Employees with most Binder Prime units sold

Emp_Num	Emp_FName	Emp_LName	Emp_Email	Total
84134	ROSALIE	GARLAND	G.ROSALI98@LGCOMPANY.COM	23
83850	RUSTY	MILES	M.RUSTY95@LGCOMPANY.COM	23



The Binder Prime Company wants to recognize the employee who sold the most of their products during a specified period.
Figure P8.23 Employees with most Binder Prime units sold

Emp_Num	Emp_FName	Emp_LName	Emp_Email	Total
84134	ROSALIE	GARLAND	G.ROSALIE@LGCOMPANY.COM	23
83850	RUSTY	MILES	M.RUSTY95@LGCOMPANY.COM	23

```

SELECT emp.emp_num, emp_fname, emp_lname, emp_email, total
FROM lgemployee emp JOIN
    (SELECT employee_id, Sum(line_qty) AS total
    FROM lginvoice i join lgline l ON i.inv_num = l.inv_num
    JOIN lgproduct p ON l.prod_sku = p.prod_sku
    JOIN lgbrand b ON b.brand_id = p.brand_id
    WHERE brand_name = 'BINDER PRIME'
    AND INV_DATE BETWEEN '01-NOV-15' AND '06-DEC-15'
    GROUP BY employee_id) sub
    ON emp.emp_num = sub.employee_id
WHERE total = (SELECT Max(total)

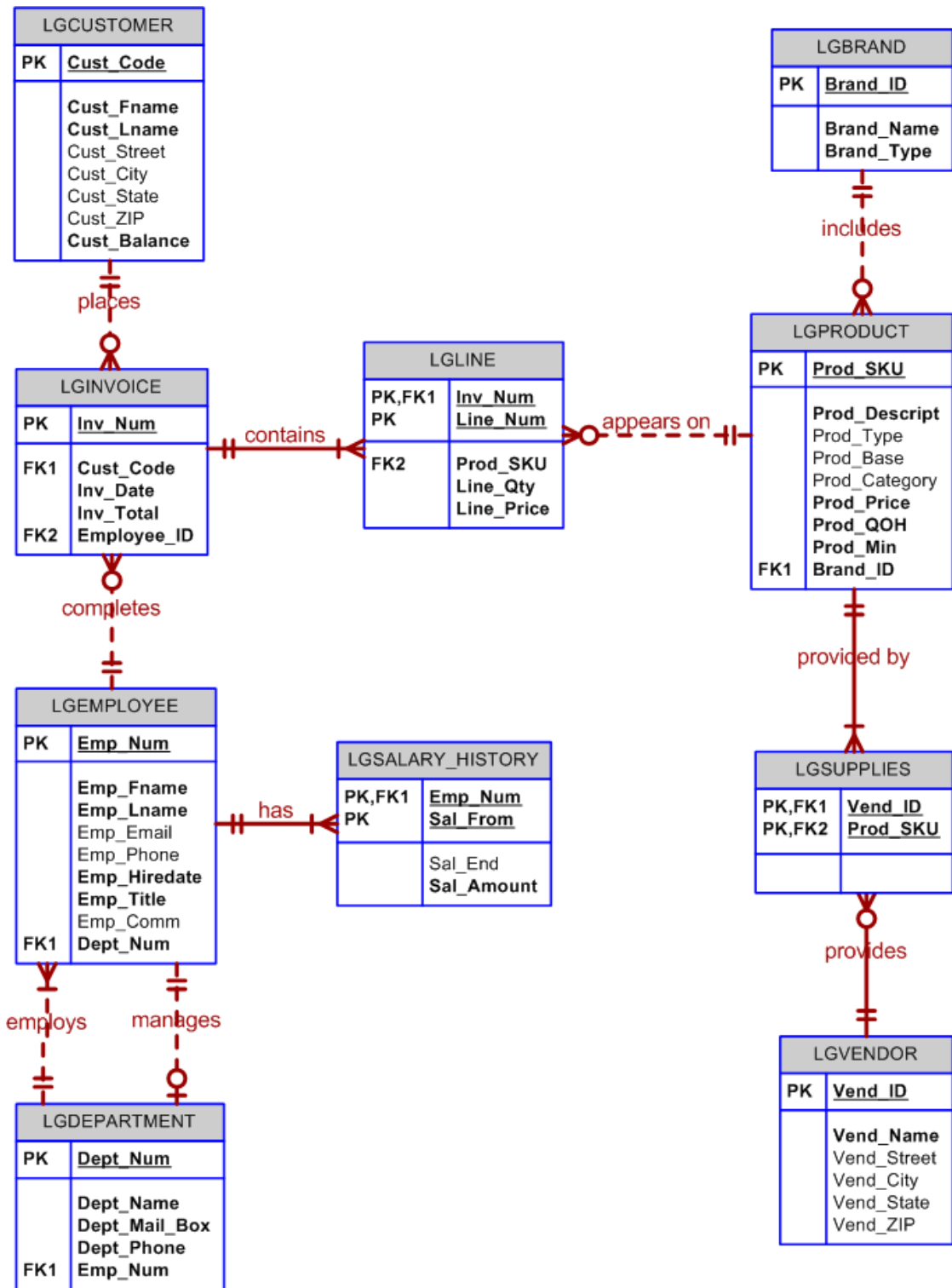
FROM (SELECT employee_id, Sum(line_qty) AS total
FROM lginvoice i JOIN lgline l ON i.inv_num = l.inv_num
JOIN lgproduct p ON l.prod_sku = p.prod_sku
JOIN lgbrand b ON b.brand_id = p.brand_id
WHERE brand_name = 'BINDER PRIME'
AND INV_DATE BETWEEN '01-NOV-15' AND '06-DEC-15'
GROUP BY employee_id));

```

•Write a query to display the customer code, first name, and last name of all customers who have had at least one invoice completed by employee 83649 and at least one invoice completed by employee 83677. Sort the output by customer last name and then first name. (Figure P8.24)

Figure P8.24 Customers with invoices filled by employees 83649 and 83677

Cust_Code	Cust_FName	Cust_LName
684	WENDI	BEAN
340	MARCIA	BURRIS
211	GERALD	CAUDILL
292	VALARIE	DILLARD
293	CLAIR	ERICKSON
416	TATIANA	HOWE
996	EZRA	LYON
98	VALENTIN	MARINO
121	PETER	SMALL
1157	LUCIO	STALEY
617	CESAR	TALLEY
457	SHAUNA	WERNER
131	SAL	WHALEY



•write a query to display the customer code, first name, and last name of all customers who have had at least one invoice col
Figure P8.24 Customers with invoices filled by employees 83649 and 83677

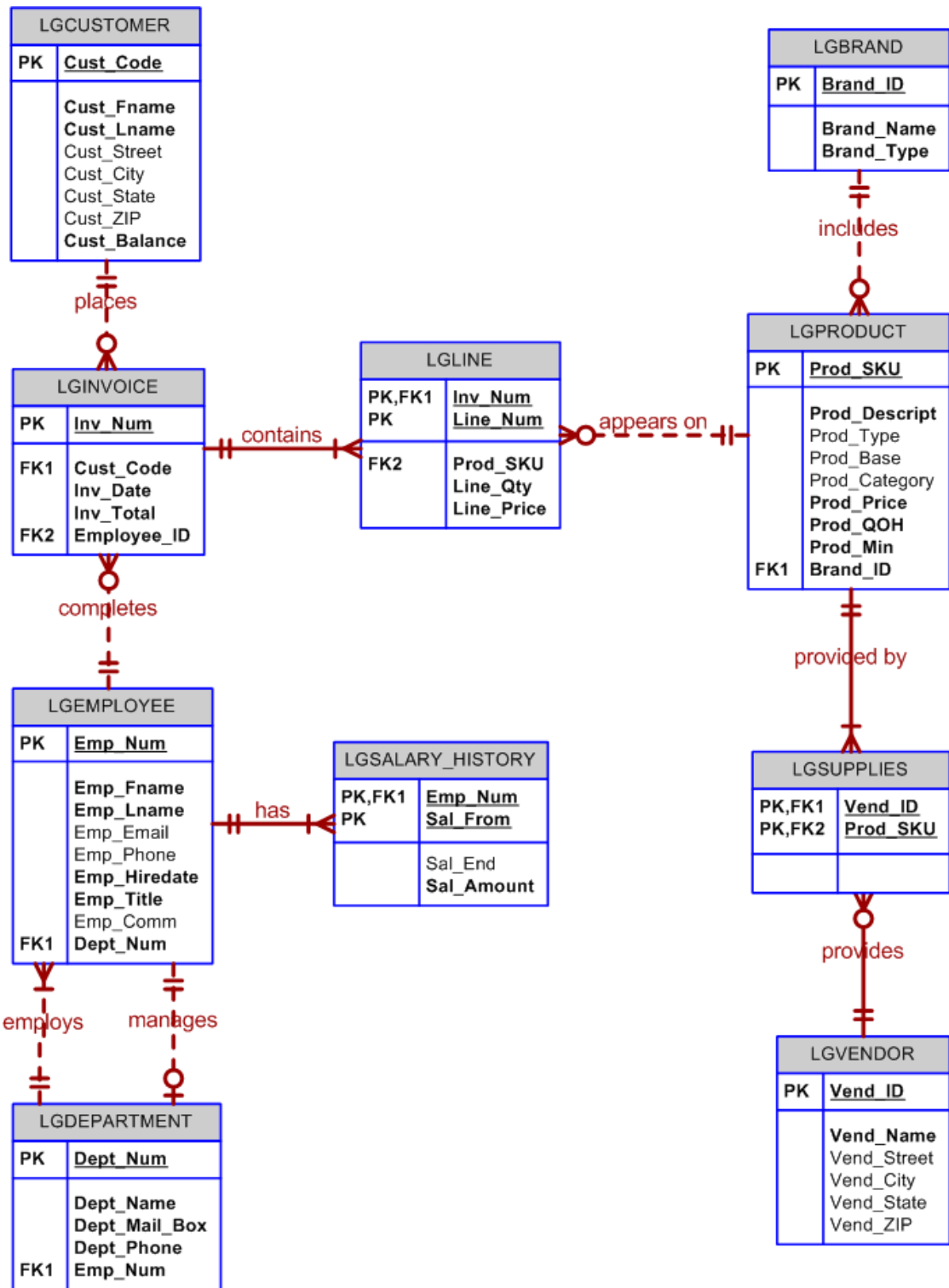
Cust_Code	Cust_FName	Cust_LName
684	WENDI	BEAN
340	MARCIA	BURRIS
211	GERALD	CAUDILL
292	VALARIE	DILLARD
293	CLAIR	ERICKSON
416	TATIANA	HOWE
996	EZRA	LYON
98	VALENTIN	MARINO
121	PETER	SMALL
1157	LUCIO	STALEY
617	CESAR	TALLEY
457	SHAUNA	WERNER
131	SAL	WHALEY

```
SELECT c.cust_code, cust_fname, cust_lname
FROM lgcustomer c JOIN lginvoice i ON c.cust_code = i.cust_code
      WHERE employee_id = 83649
INTERSECT
SELECT c.cust_code, cust_fname, cust_lname
FROM lgcustomer c JOIN lginvoice i ON c.cust_code = i.cust_code
WHERE employee_id = 83677
ORDER BY cust_lname, cust_fname;
```

•LargeCo is planning a new promotion in Alabama (AL) and wants to know about the largest purchases made by customers in that state. Write a query to display the customer code, customer first name, last name, full address, invoice date, and invoice total of the largest purchase made by each customer in Alabama. Be certain to include any customers in Alabama who have never made a purchase (their invoice dates should be NULL and the invoice totals should display as 0). (Figure P8.25)

Figure P8.25 Largest purchases of customers in Alabama

Cust_Code	Cust_FName	Cust_LName	Cust_Street	Cust_City	Cust_State	Cust_ZIP	Inv_Date	Largest Invoice
903	ROBIN	ADDISON	323 LORETTA PLACE	Mobile	AL	36693	8/26/2015	230.63
643	NINA	ALLEN	680 RED TALON DRIVE	Robertsdale	AL	36574	6/21/2015	11.99
295	DORTHY	AUSTIN	829 BIG BEND LOOP	Diamond Shamrock	AL	36614	4/24/2015	589.75
393	FOSTER	BERNAL	1299 EAST 3RD AVENUE	Birmingham	AL	35280		0
853	GAYLORD	BOLTON	1069 LUGENE LANE	Montgomery	AL	36131	11/25/2015	372.68
925	ALANA	BOOKER	1874 I STREET	Mccullough	AL	36502	12/12/2015	208.85
1248	LISA	BRADY	491 LOWLAND AVENUE	Daphne	AL	36577	12/5/2015	414.47
538	CHIKUITA	CALDWELL	1501 BRIGGS COURT	Normal	AL	35762	5/26/2015	143.9
89	MONICA	CANTRELL	697 ADAK CIRCLE	Loachapoka	AL	36865	3/31/2015	516.58
1233	NATHALIE	CHURCH	1802 SNOWY OWL CIRCLE	Napier Field	AL	36303	11/24/2015	160.96
304	GERTRUDE	CONNORS	1042 PLEASANT DRIVE	Georgiana	AL	36033	12/29/2015	376.32
1131	CARMA	CORNETT	767 CHISANA WAY	Killen	AL	35645	10/25/2015	265.12
1407	FELICIA	CRUZ	643 TURNAGAIN PARKWAY	Coalburg	AL	35068	1/6/2016	387.93



•LargeCO is planning a new promotion in Alabama (AL) and wants to know about the largest purchases made by customers in Alabama

Figure P8.25 Largest purchases of customers in Alabama

Cust_Code	Cust_FName	Cust_LName	Cust_Street	Cust_City	Cust_State	Cust_ZIP	Inv_Date	Largest Invoice
903	ROBIN	ADDISON	323 LORETTA PLACE	Mobile	AL	36693	8/26/2015	230.63
643	NINA	ALLEN	680 RED TALON DRIVE	Robertsdale	AL	36574	6/21/2015	11.99
295	DORTHY	AUSTIN	829 BIG BEND LOOP	Diamond Shamrock	AL	36614	4/24/2015	589.75
393	FOSTER	BERNAL	1299 EAST 3RD AVENUE	Birmingham	AL	35280		0
853	GAYLORD	BOLTON	1069 LUGENE LANE	Montgomery	AL	36131	11/25/2015	372.68
925	ALANA	BOOKER	1874 I STREET	Mccullough	AL	36502	12/12/2015	208.85
1248	LISA	BRADY	491 LOWLAND AVENUE	Daphne	AL	36577	12/5/2015	414.47
538	CHIKUITA	CALDWELL	1501 BRIGGS COURT	Normal	AL	35762	5/26/2015	143.9
89	MONICA	CANTRELL	697 ADAK CIRCLE	Loachapoka	AL	36865	3/31/2015	516.58
1233	NATHALIE	CHURCH	1802 SNOWY OWL CIRCLE	Napier Field	AL	36303	11/24/2015	160.96
304	GERTRUDE	CONNORS	1042 PLEASANT DRIVE	Georgiana	AL	36033	12/29/2015	376.32
1131	CARMA	CORNETT	767 CHISANA WAY	Killen	AL	35645	10/25/2015	265.12
1407	FELICIA	CRUZ	643 TURNAGAIN PARKWAY	Coalburg	AL	35068	1/6/2016	387.93

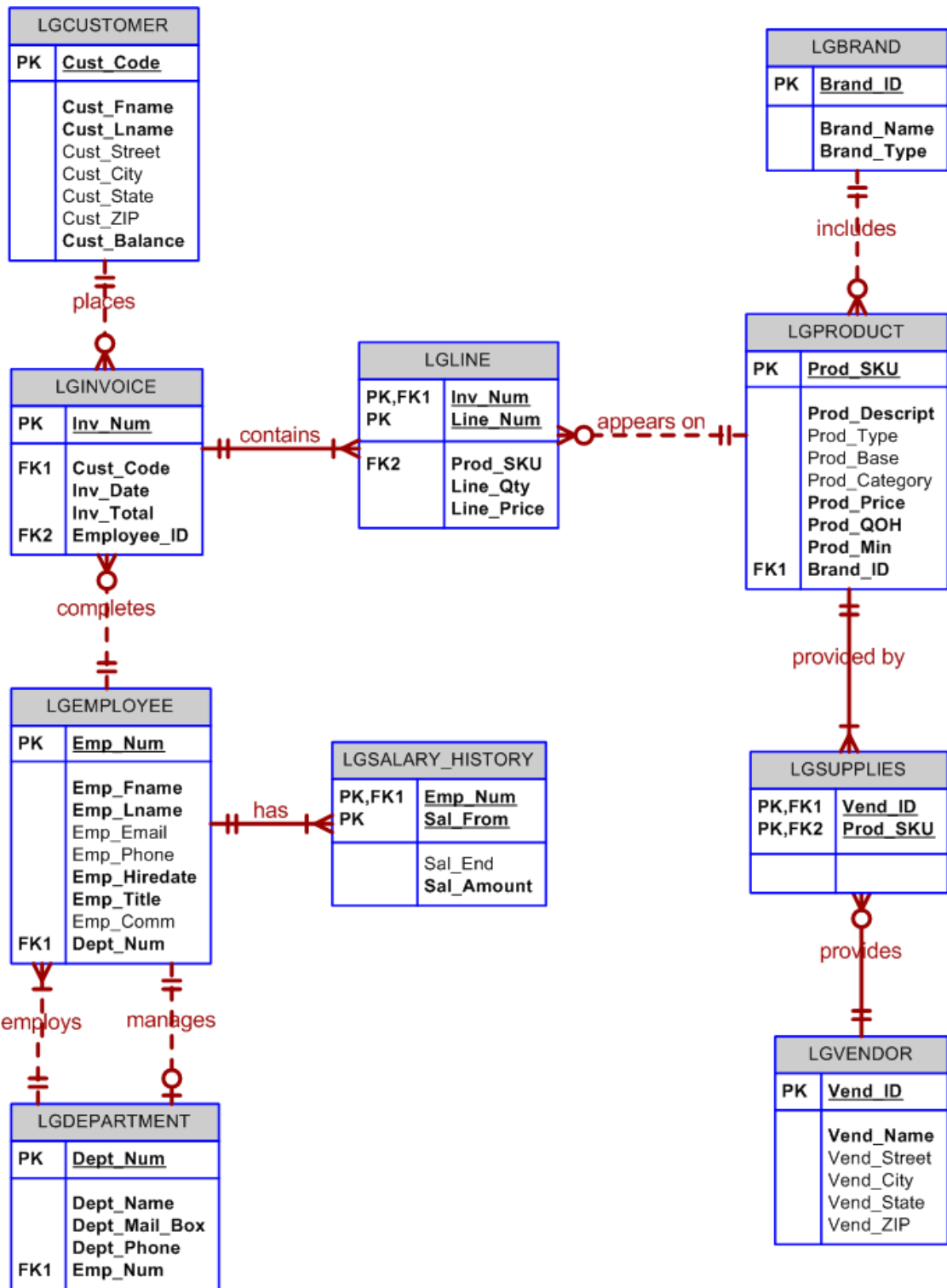
```

SELECT c.cust_code, cust_fname, cust_lname, cust_street,
cust_city, cust_state, cust_zip,
       inv_date, inv_total AS "Largest Invoice"
FROM lgcustomer c JOIN lginvoice i ON c.cust_code = i.cust_code
WHERE cust_state = 'AL'
       AND inv_total = (SELECT Max(inv_total)
                        FROM lginvoice i2
                        WHERE i2.cust_code = c.cust_code)
UNION
SELECT cust_code, cust_fname, cust_lname, cust_street, cust_city,
cust_state, cust_zip, NULL, 0
FROM lgcustomer
WHERE cust_state = 'AL'
       AND cust_code NOT IN (SELECT cust_code FROM lginvoice)
ORDER BY cust_lname, cust_fname;
```

•One of the purchasing managers is interested in the impact of product prices on the sale of products of each brand. Write a query to display the brand name, brand type, average price of products of each brand, and total units sold of products of each brand. Even if a product has been sold more than once, its price should only be included once in the calculation of the average price. However, you must be careful because multiple products of the same brand can have the same price, and each of those products must be included in the calculation of the brand's average price. (Figure P8.26)

Figure P8.26 Average price and total units sold of products by brand

Brand_Name	Brand_Type	Average Price	Units Sold
BINDER PRIME	PREMIUM	16.12	3753
BUSTERS	VALUE	22.59	3727
FORESTERS BEST	VALUE	20.94	2086
HOME COMFORT	CONTRACTOR	21.8	4842
LE MODE	PREMIUM	19.22	5284
LONG HAUL	CONTRACTOR	20.12	5728
OLDE TYME QUALITY	CONTRACTOR	18.33	3614
STUTTENFURST	CONTRACTOR	16.47	3671
VALU-MATTE	VALUE	16.84	2485



One of the purchasing managers is interested in the impact of product prices on the sale of products of each brand. Write a query to find the average price and total units sold of products by brand.

Figure P8.26 Average price and total units sold of products by brand

Brand_Name	Brand_Type	Average Price	Units Sold
BINDER PRIME	PREMIUM	16.12	3753
BUSTERS	VALUE	22.59	3727
FORESTERS BEST	VALUE	20.94	2086
HOME COMFORT	CONTRACTOR	21.8	4842
LE MODE	PREMIUM	19.22	5284
LONG HAUL	CONTRACTOR	20.12	5728
OLDE TYME QUALITY	CONTRACTOR	18.33	3614
STUTTENFURST	CONTRACTOR	16.47	3671
VALU-MATTE	VALUE	16.84	2485

```

SELECT brand_name, brand_type, Round(avgprice,2) AS "Average Price",
"Units Sold"
FROM lgbrand b JOIN (SELECT brand_id, Avg(prod_price) AS avgprice
FROM lgproduct
GROUP BY brand_id) sub1
ON b.brand_id = sub1.brand_id
JOIN (SELECT brand_id, Sum(line_qty) AS "Units Sold"
FROM lgproduct p JOIN lgline l ON p.prod_sku = l.prod_sku
GROUP BY brand_id) sub2
ON b.brand_id = sub2.brand_id
ORDER BY brand_name;

```

•The purchasing manager is still concerned about the impact of price on sales. Write a query to display the brand name, brand type, product SKU, product description, and price of any products that are not a premium brand, but that cost more than the most expensive premium brand products. (Figure P8.27)

Figure P8.27 Nonpremium products that are more expensive than premium products

Brand_Name	Brand_Type	Prod_Sku	Prod_Descript	Prod_Price
LONG HAUL	CONTRACTOR	1964-OUT	Fire Resistant Top Coat, for Interior Wood	78.49

The purchasing manager is still concerned about the impact of price on sales. write a query to display the brand name, brand type, product sku, product description, and product price for nonpremium products that are more expensive than premium products

Brand_Name	Brand_Type	Prod_Sku	Prod_Descript	Prod_Price
LONG HAUL	CONTRACTOR	1964-OUT	Fire Resistant Top Coat, for Interior Wood	78.49

```
SELECT brand_name, brand_type, prod_sku, prod_descript, prod_price
FROM lgproduct p JOIN lgbrand b ON p.brand_id = b.brand_id
WHERE brand_type <> 'PREMIUM'
AND prod_price > (SELECT Max(prod_price)
FROM lgproduct p JOIN lgbrand b ON p.brand_id = b.brand_id
WHERE brand_type = 'PREMIUM');
```

Q1. Here is a table for recording guests' stays at a hotel (arrDate denotes arrival date, depDate is departure date):

```
CREATE TABLE HotelStays
(roomNum INTEGER NOT NULL,
arrDate DATE NOT NULL,
depDate DATE NOT NULL,
guestName CHAR(30) NOT NULL,
PRIMARY KEY (roomNum, arrDate));
```

There are two problems (issues) with the above. First, the arrival date could be incorrectly entered to be later than the departure date. Second, a new entry (for a new guest) could be accidentally put in for a room number, even before the existing guest in that room has checked out:

How would you redesign the table to fix both these issues? For your answer, you can either provide a textual explanation, and/or provide SQL statements. Hint - "do not be concerned with efficiency" - ANY working solution is acceptable :)

RoomNumber Date GuestName

```
CREATE TABLE HotelStays
(roomNum INTEGER NOT NULL,
date DATE NOT NULL,
guestName CHAR(30) NOT NULL,
PRIMARY KEY (roomNum, date));
```

In other words, create a distinct row for each day a guest stays in a room! That will avoid duplicates (overlaps), and get rid of the problem of end<start.

```

CREATE OR REPLACE FUNCTION date_validator() RETURNS TRIGGER AS $exe$
BEGIN
    IF NEW.arrDate > NEW.depDate Then
        delete from HotelStays where (roomNum = NEW.roomNum AND arrDate =
NEW.arrDate);
    ELSEIF EXISTS(select arrDate, depDate from HotelStays where ((NEW.arrDate >arrDate
AND NEW.arrDate<depDate)OR (NEW.depDate >arrDate AND NEW.depDate<depDate)OR
(NEW.depDate >depDate AND NEW.arrDate<arrDate))) Then
        delete from HotelStays where (roomNum = NEW.roomNum AND arrDate =
NEW.arrDate);
    END IF;
    RETURN NEW;
END;
$exe$
Language plpgsql;

```

```

create trigger date_valid
AFTER INSERT ON HotelStays
FOR EACH ROW EXECUTE PROCEDURE date_validator();

```

```

INSERT INTO HotelStays VALUES (123, to_date('20160202', 'YYYYMMDD'),
to_date('20160206','YYYYMMDD'), 'A');
INSERT INTO HotelStays VALUES (123, to_date('20160204', 'YYYYMMDD'),

```