**Do the Exercises for Premiere Products 1a-d, 2a-d, 3a-d, 6a-d, 7, 8a-b, and 9 of the Guide to MySQL textbook, Chapter 7 on pages 208 and 209. The answer to each exercise should consist of the SQL code followed by the result. Use an appropriate SHOW command if no result set is returned.**

* 1. CREATE VIEW MAJOR\_CUSTOMER AS

SELECT CUSTOMER\_NUM, CUSTOMER\_NAME, BALANCE, CREDIT\_LIMIT, REP\_NUM

FROM CUSTOMER

WHERE CREDIT\_LIMIT <= 10000

;

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| CUSTOMER\_NUM | CUSTOMER\_NAME | BALANCE | CREDIT\_LIMIT | REP\_NUM |
| 148 | Al's Appliance and Sport | 6550.00 | 7500.00 | 20 |
| 282 | Brookings Direct | 431.50 | 10000.00 | 35 |
| 356 | Ferguson's | 5785.00 | 7500.00 | 65 |
| 408 | The Everything Shop | 5285.25 | 5000.00 | 35 |
| 462 | Bargains Galore | 3412.00 | 10000.00 | 65 |
| 608 | Johnson's Department Store | 2106.00 | 10000.00 | 65 |
| 687 | Lee's Sport and Appliance | 2851.00 | 5000.00 | 35 |
| 725 | Deerfield's Four Seasons | 248.00 | 7500.00 | 35 |
| 842 | All Season | 8221.00 | 7500.00 | 20 |

* 1. SELECT CUSTOMER\_NUM, CUSTOMER\_NAME

FROM MAJOR\_CUSTOMER

WHERE BALANCE > CREDIT\_LIMIT

;

|  |  |
| --- | --- |
| CUSTOMER\_NUM | CUSTOMER\_NAME |
| 408 | The Everything Shop |
| 842 | All Season |

* 1. SELECT CUSTOMER\_NUM, CUSTOMER\_NAME

FROM CUSTOMER

WHERE BALANCE > CREDIT\_LIMIT

;

|  |  |
| --- | --- |
| CUSTOMER\_NUM | CUSTOMER\_NAME |
| 408 | The Everything Shop |
| 842 | All Season |

* 1. Updating the database through this view should cause no significant problems as the omitted fields accept nulls and the only data not displayed is the customer’s address.
  2. CREATE VIEW PART\_ORDER AS

SELECT OL.PART\_NUM, P.DESCRIPTION, P.PRICE, OL.ORDER\_NUM, O.ORDER\_DATE, OL.NUM\_ORDERED, OL.QUOTED\_PRICE

FROM ORDER\_LINE OL, PART P, ORDERS O

WHERE OL.PART\_NUM = P.PART\_NUM AND O.ORDER\_NUM = OL.ORDER\_NUM

;

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| PART\_NUM | DESCRIPTION | PRICE | ORDER\_NUM | ORDER\_DATE | NUM\_ORDERED | QUOTED\_PRICE |
| AT94 | Iron | 24.95 | 21608 | 2007-10-20 | 11 | 21.95 |
| DR93 | Gas Range | 495.00 | 21610 | 2007-10-20 | 1 | 495.00 |
| DW11 | Washer | 399.99 | 21610 | 2007-10-20 | 1 | 399.99 |
| KL62 | Dryer | 349.95 | 21613 | 2007-10-21 | 4 | 329.95 |
| KT03 | Dishwasher | 595.00 | 21614 | 2007-10-21 | 2 | 595.00 |
| BV06 | Home Gym | 794.95 | 21617 | 2007-10-23 | 2 | 794.95 |
| CD52 | Microwave Oven | 165.00 | 21617 | 2007-10-23 | 4 | 150.00 |
| DR93 | Gas Range | 495.00 | 21619 | 2007-10-23 | 1 | 495.00 |
| KV29 | Treadmill | 1390.00 | 21623 | 2007-10-23 | 2 | 1290.00 |

* 1. SELECT PART\_NUM, DESCRIPTION, ORDER\_NUM, QUOTED\_PRICE

FROM PART\_ORDER

WHERE QUOTED\_PRICE > 100

;

|  |  |  |  |
| --- | --- | --- | --- |
| PART\_NUM | DESCRIPTION | ORDER\_NUM | QUOTED\_PRICE |
| DR93 | Gas Range | 21610 | 495.00 |
| DW11 | Washer | 21610 | 399.99 |
| KL62 | Dryer | 21613 | 329.95 |
| KT03 | Dishwasher | 21614 | 595.00 |
| BV06 | Home Gym | 21617 | 794.95 |
| CD52 | Microwave Oven | 21617 | 150.00 |
| DR93 | Gas Range | 21619 | 495.00 |
| KV29 | Treadmill | 21623 | 1290.00 |

* 1. SELECT OL.PART\_NUM, P.DESCRIPTION, OL.ORDER\_NUM, OL.QUOTED\_PRICE

FROM ORDER\_LINE OL, PART P

WHERE OL.PART\_NUM = P.PART\_NUM AND QUOTED\_PRICE > 100

;

|  |  |  |  |
| --- | --- | --- | --- |
| PART\_NUM | DESCRIPTION | ORDER\_NUM | QUOTED\_PRICE |
| DR93 | Gas Range | 21610 | 495.00 |
| DW11 | Washer | 21610 | 399.99 |
| KL62 | Dryer | 21613 | 329.95 |
| KT03 | Dishwasher | 21614 | 595.00 |
| BV06 | Home Gym | 21617 | 794.95 |
| CD52 | Microwave Oven | 21617 | 150.00 |
| DR93 | Gas Range | 21619 | 495.00 |
| KV29 | Treadmill | 21623 | 1290.00 |

* 1. There could potentially be some problems while updating the database through this view as not all of the fields used will accept a null value and the values span 3 separate tables. This could make updating the database through this view quite difficult for the individual using it or result in a loss of data integrity in the tables used.
  2. CREATE VIEW ORDER\_TOTAL AS

SELECT ORDER\_NUM, SUM(NUM\_ORDERED \* QUOTED\_PRICE) AS TOTAL\_AMOUNT

FROM ORDER\_LINE

GROUP BY ORDER\_NUM

ORDER BY ORDER\_NUM

;

|  |  |
| --- | --- |
| ORDER\_NUM | TOTAL\_AMOUNT |
| 21608 | 241.45 |
| 21610 | 894.99 |
| 21613 | 1319.80 |
| 21614 | 1190.00 |
| 21617 | 2189.90 |
| 21619 | 495.00 |
| 21623 | 2580.00 |

* 1. SELECT ORDER\_NUM, TOTAL\_AMOUNT

FROM ORDER\_TOTAL

WHERE TOTAL\_AMOUNT > 1000

;

|  |  |
| --- | --- |
| ORDER\_NUM | TOTAL\_AMOUNT |
| 21613 | 1319.80 |
| 21614 | 1190.00 |
| 21617 | 2189.90 |
| 21623 | 2580.00 |

* 1. SELECT ORDER\_NUM, SUM(NUM\_ORDERED \* QUOTED\_PRICE) AS TOTAL\_AMOUNT

FROM ORDER\_LINE

GROUP BY ORDER\_NUM

HAVING SUM(NUM\_ORDERED \* QUOTED\_PRICE) > 1000

ORDER BY ORDER\_NUM

;

|  |  |
| --- | --- |
| ORDER\_NUM | TOTAL\_AMOUNT |
| 21613 | 1319.80 |
| 21614 | 1190.00 |
| 21617 | 2189.90 |
| 21623 | 2580.00 |

* 1. Updating the database through this view has the potential to cause quite a few problems. One problem is because there is a calculated field in the view that would not be able to be updated directly because that column does not exist in the base table. Additionally, the ORDER\_NUM field is grouped to into single rows which could cause problems with conflicting data or wrong data should a change be attempted.
  2. CREATE INDEX PART\_INDEX1 ON ORDER\_LINE(PART\_NUM);

SHOW INDEX FROM ORDER\_LINE;

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Table | Non\_unique | Key\_name | Seq\_in\_index | Column\_name | Collation | Cardinality | Sub\_part | Packed | Null | Index\_type | Comment | Index\_comment |
| order\_line | 0 | PRIMARY | 1 | ORDER\_NUM | A | 7 | NULL | NULL |  | BTREE |  |  |
| order\_line | 0 | PRIMARY | 2 | PART\_NUM | A | 9 | NULL | NULL |  | BTREE |  |  |
| order\_line | 1 | PART\_INDEX1 | 1 | PART\_NUM | A | 8 | NULL | NULL |  | BTREE |  |  |

* 1. CREATE INDEX PART\_INDEX2 ON PART(CLASS);

SHOW INDEX FROM PART;

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Table | Non\_unique | Key\_name | Seq\_in\_index | Column\_name | Collation | Cardinality | Sub\_part | Packed | Null | Index\_type | Comment | Index\_comment |
| part | 0 | PRIMARY | 1 | PART\_NUM | A | 10 | NULL | NULL |  | BTREE |  |  |
| part | 1 | PART\_INDEX2 | 1 | CLASS | A | 3 | NULL | NULL | YES | BTREE |  |  |

* 1. CREATE INDEX PART\_INDEX3 ON PART(CLASS, WAREHOUSE);

SHOW INDEX FROM PART;

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Table | Non\_unique | Key\_name | Seq\_in\_index | Column\_name | Collation | Cardinality | Sub\_part | Packed | Null | Index\_type | Comment | Index\_comment |
| part | 0 | PRIMARY | 1 | PART\_NUM | A | 10 | NULL | NULL |  | BTREE |  |  |
| part | 1 | PART\_INDEX2 | 1 | CLASS | A | 3 | NULL | NULL | YES | BTREE |  |  |
| part | 1 | PART\_INDEX3 | 1 | CLASS | A | 3 | NULL | NULL | YES | BTREE |  |  |
| part | 1 | PART\_INDEX3 | 2 | WAREHOUSE | A | 5 | NULL | NULL | YES | BTREE |  |  |

* 1. CREATE INDEX PART\_INDEX4 ON PART(CLASS DESC, WAREHOUSE);

SHOW INDEX FROM PART;

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Table | Non\_unique | Key\_name | Seq\_in\_index | Column\_name | Collation | Cardinality | Sub\_part | Packed | Null | Index\_type | Comment | Index\_comment |
| part | 0 | PRIMARY | 1 | PART\_NUM | A | 10 | NULL | NULL |  | BTREE |  |  |
| part | 1 | PART\_INDEX2 | 1 | CLASS | A | 3 | NULL | NULL | YES | BTREE |  |  |
| part | 1 | PART\_INDEX3 | 1 | CLASS | A | 3 | NULL | NULL | YES | BTREE |  |  |
| part | 1 | PART\_INDEX3 | 2 | WAREHOUSE | A | 5 | NULL | NULL | YES | BTREE |  |  |
| part | 1 | PART\_INDEX4 | 1 | CLASS | A | 3 | NULL | NULL | YES | BTREE |  |  |
| part | 1 | PART\_INDEX4 | 2 | WAREHOUSE | A | 5 | NULL | NULL | YES | BTREE |  |  |

1. DROP INDEX PART\_INDEX3 ON PART;

SHOW INDEX FROM PART;

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Table | Non\_unique | Key\_name | Seq\_in\_index | Column\_name | Collation | Cardinality | Sub\_part | Packed | Null | Index\_type | Comment | Index\_comment |
| part | 0 | PRIMARY | 1 | PART\_NUM | A | 10 | NULL | NULL |  | BTREE |  |  |
| part | 1 | PART\_INDEX2 | 1 | CLASS | A | 3 | NULL | NULL | YES | BTREE |  |  |
| part | 1 | PART\_INDEX4 | 1 | CLASS | A | 3 | NULL | NULL | YES | BTREE |  |  |
| part | 1 | PART\_INDEX4 | 2 | WAREHOUSE | A | 5 | NULL | NULL | YES | BTREE |  |  |

* 1. SHOW TABLES;

|  |
| --- |
| Tables\_in\_premiere |
| customer |
| major\_customer |
| order\_line |
| order\_total |
| orders |
| part |
| part\_order |
| rep |

* 1. SHOW COLUMNS FROM PART;

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Type | Null | Key | Default | Extra |
| PART\_NUM | char(4) | NO | PRI | NULL |  |
| DESCRIPTION | char(15) | YES |  | NULL |  |
| ON\_HAND | decimal(4,0) | YES |  | NULL |  |
| CLASS | char(2) | YES | MUL | NULL |  |
| WAREHOUSE | char(1) | YES |  | NULL |  |
| PRICE | decimal(6,2) | YES |  | NULL |  |

1. ALTER TABLE ORDER\_LINE

ADD FOREIGN KEY (ORDER\_NUM) REFERENCES ORDERS(ORDER\_NUM)

;