Nathan Hamilton

CSCD 327-01

Lab 5

1. SELECT Title, Retail

FROM BOOKS

WHERE Retail < (SELECT avg(Retail)

FROM BOOKS);

|  |  |
| --- | --- |
| HOW TO GET FASTER PIZZA | 29.95 |
| THE WOK WAY TO COOK | 28.75 |
| REVENGE OF MICKEY | 22.00 |
| BODYBUILD IN 10 MINUTES A DAY | 30.95 |
| HANDCRANKED COMPUTERS | 25.00 |
| SHORTEST POEMS | 39.95 |
| COOKING WITH MUSHROOMS | 19.95 |
| BIG BEAR AND LITTLE DOVE | 8.95 |
| HOW TO MANAGE THE MANAGER | 31.95 |

2. SELECT Title, Category, Cost

FROM BOOKS b1

WHERE Cost < (SELECT avg(Cost)

FROM BOOKS b2

GROUP BY Category

HAVING b1.Category = b2.Category);

|  |  |  |
| --- | --- | --- |
| REVENGE OF MICKEY | FAMILY LIFE | 14.20 |
| HANDCRANKED COMPUTERS | COMPUTER | 21.80 |
| COOKING WITH MUSHROOMS | COOKING | 12.50 |
| BIG BEAR AND LITTLE DOVE | CHILDREN | 5.32 |
| DATABASE IMPLEMENTATION | COMPUTER | 31.40 |

3. SELECT Order\_num

FROM ORDERS

WHERE ShipState = (SELECT ShipState

FROM ORDERS

WHERE Order\_num = 1014);

|  |
| --- |
| 1007 |
| 1014 |
|  |

4. SELECT Order\_num, sum(Quantity \* PaidEach)

FROM ORDERITEMS

GROUP BY Order\_num

Having sum(Quantity \* PaidEach) > (SELECT sum(Quantity \* PaidEach)

FROM ORDERITEMS

WHERE Order\_num = 1002);

|  |  |
| --- | --- |
| 1001 | 117.40 |
| 1004 | 170.90 |
| 1007 | 335.85 |
| 1012 | 166.40 |

5. SELECT Title

FROM BOOKS

WHERE Category IN (SELECT Category

FROM BOOKS NATURAL JOIN ORDERITEMS NATURAL JOIN ORDERS

WHERE Customer\_num = 1007)

AND Title NOT IN (SELECT Title

FROM BOOKS NATURAL JOIN ORDERITEMS NATURAL JOIN ORDERS

WHERE Customer\_num = 1007);

|  |
| --- |
| HANDCRANKED COMPUTERS |
| PAINLESS CHILD-REARING |
| BUILDING A CAR WITH TOOTHPICKS |

6. SELECT ShipCity, ShipState

FROM ORDERS

WHERE datediff(ShipDate, OrderDate) = (SELECT max(datediff(ShipDate, OrderDate))

FROM ORDERS);

|  |  |
| --- | --- |
| TRENTON | NJ |

7. SELECT Customer\_num

FROM ORDERS NATURAL JOIN ORDERITEMS NATURAL JOIN BOOKS

WHERE Retail = (SELECT min(retail)

FROM BOOKS);

|  |
| --- |
| 1007 |
| 1017 |
| 1015 |

8. SELECT count(DISTINCT Customer\_num)

FROM (SELECT Customer\_num

FROM ORDERS NATURAL JOIN ORDERITEMS NATURAL JOIN BOOKS NATURAL JOIN

BOOKAUTHOR NATURAL JOIN AUTHOR

WHERE Fname = 'JAMES' AND Lname = 'AUSTIN') TEMP;

5

9. SELECT product\_name, list\_price

FROM products

WHERE list\_price > (SELECT avg(list\_price)

FROM products)

ORDER BY list\_price DESC;

|  |  |
| --- | --- |
| Gibson SG | 2517.00 |
| Gibson Les Paul | 1199.00 |

10. SELECT category\_name

FROM categories c1

WHERE NOT EXISTS (SELECT \*

FROM categories NATURAL JOIN products c2

WHERE c1.category\_id = c2.category\_id);

Keyboards

11. SELECT email\_address, order\_id, order\_date

FROM customers NATURAL JOIN orders t1

WHERE order\_date = (SELECT min(order\_date)

FROM customers NATURAL JOIN orders t2

WHERE t1.customer\_id = t2.customer\_id);

|  |  |  |
| --- | --- | --- |
| allan.sherwood@yahoo.com | 1 | 2012-03-28 09:40:28 |
| barryz@gmail.com | 2 | 2012-03-28 11:23:20 |
| christineb@solarone.com | 4 | 2012-03-30 15:22:31 |
| david.goldstein@hotmail.com | 5 | 2012-03-31 05:43:11 |
| erinv@gmail.com | 6 | 2012-03-31 18:37:22 |
| frankwilson@sbcglobal.net | 7 | 2012-04-01 23:11:12 |
| gary\_hernandez@yahoo.com | 8 | 2012-04-02 11:26:38 |

12. SELECT ID, name, salary

FROM instructor

WHERE salary = (SELECT max(salary)

FROM instructor);

|  |  |  |
| --- | --- | --- |
| 22222 | Einstein | 95000.00 |
|  |  |  |

13. SELECT i1.ID, i1.name, i1.salary

FROM instructor i1

WHERE i1.salary = (SELECT max(i2.salary)

FROM instructor i2

WHERE i2.salary < (SELECT max(i3.salary)

FROM instructor i3));

|  |  |  |
| --- | --- | --- |
| 83821 | Brandt | 92000.00 |
|  |  |  |

14. SELECT ID, name, count(DISTINCT course\_id)

FROM instructor NATURAL JOIN teaches t1

GROUP BY t1.ID

HAVING count(DISTINCT course\_id) >= ALL (SELECT count(DISTINCT course\_id)

FROM instructor NAUTAL JOIN teaches t2

GROUP BY t2.ID);

|  |  |  |
| --- | --- | --- |
| 10101 | Srinivasan | 3 |