|  |  |
| --- | --- |
| **Short database description "Computer firm":**  The database scheme consists of four tables:  Product(maker, model, type) PC(code, model, speed, ram, hd, cd, price) Laptop(code, model, speed, ram, hd, screen, price) Printer(code, model, color, type, price) The Product table contains data on the maker, model number, and type of product ('PC', 'Laptop', or 'Printer'). It is assumed that model numbers in the Product table are unique for all makers and product types. Each personal computer in the PC table is unambiguously identified by a unique code, and is additionally characterized by its model (foreign key referring to the Product table), processor speed (in MHz) – speed field, RAM capacity (in Mb) - ram, hard disk drive capacity (in Gb) – hd, CD-ROM speed (e.g, '4x') - cd, and its price. The Laptop table is similar to the PC table, except that instead of the CD-ROM speed, it contains the screen size (in inches) – screen. For each printer model in the Printer table, its output type (‘y’ for color and ‘n’ for monochrome) – color field, printing technology ('Laser', 'Jet', or 'Matrix') – type, and price are specified.  **Short database description "Recycling firm":**  The firm owns several buy-back centers for collection of recyclable materials. Each of them receives funds to be paid to the recyclables suppliers. Data on funds received is recorded in the table Income\_o(point, date, inc) The primary key is (point, date), where point holds the identifier of the buy-back center, and date corresponds to the calendar date the funds were received. The date column doesn’t include the time part, thus, money (inc) arrives no more than once a day for each center. Information on payments to the recyclables suppliers is held in the table Outcome\_o(point, date, out) In this table, the primary key (point, date) ensures each buy-back center reports about payments (out) no more than once a day, too. For the case income and expenditure may occur more than once a day, another database schema with tables having a primary key consisting of the single column code is used: Income(code, point, date, inc) Outcome(code, point, date, out) Here, the date column doesn’t include the time part, either. | **Short database description "Ships":**  The database of naval ships that took part in World War II is under consideration. The database consists of the following relations: Classes(class, type, country, numGuns, bore, displacement) Ships(name, class, launched) Battles(name, date) Outcomes(ship, battle, result) Ships in classes all have the same general design. A class is normally assigned either the name of the first ship built according to the corresponding design, or a name that is different from any ship name in the database. The ship whose name is assigned to a class is called a lead ship. The Classes relation includes the name of the class, type (can be either bb for a battle ship, or bc for a battle cruiser), country the ship was built in, the number of main guns, gun caliber (bore diameter in inches), and displacement (weight in tons). The Ships relation holds information about the ship name, the name of its corresponding class, and the year the ship was launched. The Battles relation contains names and dates of battles the ships participated in, and the Outcomes relation - the battle result for a given ship (may be sunk, damaged, or OK, the last value meaning the ship survived the battle unharmed). Notes: 1) The Outcomes relation may contain ships not present in the Ships relation. 2) A ship sunk can’t participate in later battles. 3) For historical reasons, lead ships are referred to as head ships in many exercises.4) A ship found in the Outcomes table but not in the Ships table is still considered in the database. This is true even if it is sunk. |
| **Exercise** | **My result** |
| Find the model number, speed and hard drive capacity for all the PCs with prices below $500. Result set: model, speed, hd. | SELECT model, speed, hd  FROM PC  where price < 500 |
| List all printer makers. Result set: maker. | SELECT DISTINCT maker  FROM Product  WHERE type = 'printer' |
| Find the model number, RAM and screen size of the laptops with prices over $1000. | SELECT model, ram, screen  FROM Laptop  WHERE price > 1000 |
| Find all records from the Printer table containing data about color printers. | Select \*  FROM Printer  WHERE color = 'y' |
| Find the model number, speed and hard drive capacity of PCs cheaper than $600 having a 12x or a 24x CD drive. | Select model, speed, hd  FROM PC  WHERE price < 600 AND (cd = '12x' OR cd = '24x') |
| For each maker producing laptops with a hard drive capacity of 10 Gb or higher, find the speed of such laptops. Result set: maker, speed. | Select DISTINCT maker, speed  FROM PRODUCT s  JOIN Laptop d ON s.model = d.model  where hd >= 10  ORDER BY maker |
| Find the makers of PCs with a processor speed of 450 MHz or more. Result set: maker. | SELECT DISTINCT Maker  FROM PRODUCT  JOIN PC ON PC.model = PRODUCT.model  WHERE speed >= 450 |
| Find the printer models having the highest price. Result set: model, price. | SELECT model, MAX(price) AS PRICE  FROM Printer  GROUP BY model  HAVING MAX(price) = (SELECT MAX(price) FROM PRINTER) |
| Find out the average speed of PCs. | Select AVG(speed)  FROM PC |
| Find out the average speed of the laptops priced over $1000. | SELECT AVG(speed) AS Average  FROM Laptop  WHERE price > 1000 |
| Find out the average speed of the PCs produced by maker A. | SELECT AVG(speed)  FROM PC  JOIN Product ON PC.model = Product.model  WHERE maker = 'A' |
| For the ships in the Ships table that have at least 10 guns, get the class, name, and country. | SELECT ships.class, name, country  FROM Ships  JOIN Classes ON ships.class = classes.class  WHERE numGuns >= 10  ORDER BY CLASS |
| Get hard drive capacities that are identical for two or more PCs. Result set: hd. | SELECT hd  FROM PC  GROUP BY hd  HAVING COUNT(model) >= 2 |
| For each maker having models in the Laptop table, find out the average screen size of the laptops he produces. Result set: maker, average screen size. | Select Maker, AVG(screen)  FROM Laptop  JOIN Product ON Laptop.model = Product.model  GROUP BY Maker |
| Find the makers producing at least three distinct models of PCs.  Result set: maker, number of PC models. | SELECT maker, COUNT(maker) AS Count\_Model  FROM Product  WHERE type = 'PC'  GROUP by maker  HAVING COUNT(maker) >= 3 |
| Find out the maximum PC price for each maker having models in the PC table. Result set: maker, maximum price. | SELECT maker, MAX(price) AS Maximum\_price  FROM PC  JOIN Product ON PC.model = Product.model  GROUP BY maker |
| For each value of PC speed that exceeds 600 MHz, find out the average price of PCs with identical speeds.  Result set: speed, average price. | SELECT speed, AVG(price) AS Average\_price  FROM PC  GROUP BY speed  HAVING speed > 600 |
| Get the makers producing both PCs having a speed of 750 MHz or higher and laptops with a speed of 750 MHz or higher.  Result set: maker | SELECT maker  FROM Product  JOIN PC ON Product.model = PC.model  WHERE speed >= 750  INTERSECT  SELECT maker  FROM Product  JOIN Laptop ON Product.model = Laptop.model  WHERE speed >= 750 |
| Using Product table, find out the number of makers who produce only one model. | SELECT COUNT(A.Qty)  FROM(SELECT maker, COUNT(model) AS QTY  FROM PRODUCT  GROUP by maker) A  Where A.qty = 1 |
| For ship classes with a gun caliber of 16 in. or more, display the class and the country. | SELECT class,country  FROM classes  WHERE bore >= 16 |
| Get the ships sunk in the North Atlantic battle.  Result set: ship. | Select ship  FROM outcomes  where battle = 'North Atlantic' AND result = 'sunk' |
| List the names of lead ships in the database (including the Outcomes table). | SELECT name  FROM SHIPS  where name = class  UNION  (SELECT ship  FROM Outcomes  intersect  Select class  FROM classes) |
| Find countries that ever had classes of both battleships (‘bb’) and cruisers (‘bc’). | SELECT country  FROM classes  where type = 'bb'  intersect  SELECT Country  FROM classes  where type = 'bc' |
| Find the names of ships sunk at battles, along with the names of the corresponding battles. | SELECT ship, battle  FROM Outcomes  WHERE result = 'sunk' |
| Find all ship names beginning with the letter R. | SELECT name AS ship  FROM Ships  WHERE name LIKE 'R%'  UNION  SELECT ship  FROM Outcomes  WHERE ship LIKE 'R%' |
| Find all ship names consisting of three or more words (e.g., King George V).  Consider the words in ship names to be separated by single spaces, and the ship names to have no leading or trailing spaces. | SELECT name AS ship  FROM Ships  WHERE name LIKE '%% %% %%'  UNION  SELECT ship  FROM Outcomes  WHERE ship LIKE '%% %% %%' |
| Find the ship classes having at least one ship sunk in battles. | SELECT Class  FROM Ships  JOIN Outcomes ON Ships.name = Outcomes.ship  WHERE result = 'sunk'  UNION  SELECT Class  FROM Classes  JOIN Outcomes ON Classes.class = Outcomes.ship  WHERE result = 'sunk' |
| Find the names of the ships having a gun caliber of 16 inches (including ships in the Outcomes table). | SELECT name  FROM Ships  WHERE class IN(Select class  FROM classes  WHERE bore = '16')  UNION  SELECT ship AS name  FROM Outcomes  Where ship IN (Select class  FROM classes  WHERE bore = '16') |
| Find the battles in which Kongo-class ships from the Ships table were engaged. | Select DISTINCT battle  FROM Outcomes a  Join Ships b ON a.ship = b.name  WHERE class= 'Kongo' |
| With a precision of two decimal places, determine the average number of guns for the battleship classes. | SELECT CAST(AVG(NumGuns\*1.0) AS NUMERIC(6,2))  FROM Classes  WHERE type = 'bb' |
| For each class, determine the year the first ship of this class was launched.  If the lead ship’s year of launch is not known, get the minimum year of launch for the ships of this class.  Result set: class, year. | Select a.class, MIN(b.launched) as year  FROM Classes a  LEFT JOIN SHIPS b ON a.class = b.class  GROUP BY a.class |
| For the database with money transactions being recorded not more than once a day, calculate the total cash balance of all buy-back centers at the beginning of 04/15/2001 | SELECT(Select SUM(INC)  FROM Income\_o  Where date < '20010415')  -  (Select SUM(out)  FROM Outcome\_o  Where date < '20010415') AS REMAIN |
| Find the computer equipment makers not producing any PC models absent in the PC table. | SELECT maker  FROM Product  Except  (Select DISTINCT maker  FROM product  where model IN(  SELECT model  FROM PRODUCT  where type = 'PC'  Except  Select model  From Product  where model  IN (Select model FROM PC))) |
|  |  |