**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.

|  |  |
| --- | --- |
| **Exercise: 1** | **My solution:** |
| 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 |

|  |  |
| --- | --- |
| **Exercise: 2** | **My solution:** |
| List all printer makers.  Result set: maker. | SELECT DISTINCT maker  FROM Product  WHERE type = 'Printer' |

|  |  |
| --- | --- |
| **Exercise: 3** | **My solution:** |
| Find the model number, RAM and screen size of the laptops with prices over $1000. | SELECT model, ram, screen  FROM Laptop  WHERE price > 1000 |

|  |  |
| --- | --- |
| **Exercise: 4** | **My solution:** |
| Find all records from the Printer table containing data about color printers. | SELECT code, model, color, type, price  FROM Printer  WHERE color = 'y' |

|  |  |
| --- | --- |
| **Exercise: 5** | **My solution:** |
| 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') |

|  |  |
| --- | --- |
| **Exercise: 6** | **My solution:** |
| 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 Product.maker, Laptop.speed  FROM Product  INNER JOIN Laptop ON Product.model=Laptop.model  WHERE Laptop.hd >= 10 |

|  |  |
| --- | --- |
| **Exercise: 7** | **My solution:** |
| Get the models and prices for all commercially available products (of any type) produced by maker B. | SELECT DISTINCT Product.model, PC.price  FROM PC  INNER JOIN Product ON PC.model=Product.model  WHERE Product.maker = 'B'  UNION  SELECT DISTINCT Product.model, Laptop.price  FROM Laptop  INNER JOIN Product ON Laptop.model=Product.model  WHERE Product.maker = 'B'  UNION  SELECT DISTINCT Product.model, Printer.price  FROM Printer  INNER JOIN Product ON Printer.model=Product.model  WHERE Product.maker = 'B' |

|  |  |
| --- | --- |
| **Exercise: 8** | **My solution:** |
| Find the makers producing PCs but not laptops. | SELECT COALESCE(m\_pc.maker, m\_laptop.maker) FROM  (SELECT DISTINCT maker FROM Product WHERE type='PC') m\_pc  FULL JOIN  (SELECT DISTINCT maker FROM Product WHERE type='Laptop') m\_laptop  ON m\_pc.maker = m\_laptop.maker  WHERE m\_laptop.maker IS NULL |

|  |  |
| --- | --- |
| **Exercise: 9** | **My solution:** |
| Find the makers of PCs with a processor speed of 450 MHz or more.  Result set: maker. | SELECT DISTINCT maker  FROM Product  INNER JOIN PC ON Product.model = PC.model  WHERE PC.speed >=450 |

|  |  |
| --- | --- |
| **Exercise: 10** | **My solution:** |
| Find the printer models having the highest price.  Result set: model, price. | SELECT model, price  FROM Printer  WHERE price = (SELECT MAX(price) from Printer) |

|  |  |
| --- | --- |
| **Exercise: 11** | **My solution:** |
| Find out the average speed of PCs. | SELECT AVG(speed)  FROM PC |

|  |  |
| --- | --- |
| **Exercise: 12** | **My solution:** |
| Find out the average speed of the laptops priced over $1000. | SELECT AVG(speed)  FROM Laptop  WHERE price > 1000 |

|  |  |
| --- | --- |
| **Exercise: 13** | **My solution:** |
| Find out the average speed of the PCs produced by maker A. | SELECT DISTINCT AVG(speed)  FROM PC  INNER JOIN Product ON Product.model = PC.model  WHERE Product.maker = 'A' |

|  |  |
| --- | --- |
| **Exercise: 14** | **My solution:** |
| For the ships in the Ships table that have at least 10 guns, get the class, name, and country. | SELECT Ships.class, Ships.name, Classes.country  FROM Ships  INNER JOIN Classes ON Ships.class = Classes.class  WHERE Classes.numGuns >= 10 |

|  |  |
| --- | --- |
| **Exercise: 15** | **My solution:** |
| Get hard drive capacities that are identical for two or more PCs.  Result set: hd. | SELECT hd  FROM PC  GROUP BY hd  HAVING COUNT(hd)>1 |

|  |  |
| --- | --- |
| **Exercise: 16** | **My solution:** |
| Get pairs of PC models with identical speeds and the same RAM capacity. Each resulting pair should be displayed only once, i.e. (i, j) but not (j, i).  Result set: model with the bigger number, model with the smaller number, speed, and RAM. | SELECT DISTINCT A.model AS model, B.model AS model, A.speed AS speed, A.ram AS ram  FROM PC AS A, PC B  WHERE A.ram = B.ram AND  A.speed = B.speed AND  A.model > B.model |

|  |  |
| --- | --- |
| **Exercise: 17** | **My solution:** |
| Get the laptop models that have a speed smaller than the speed of any PC.  Result set: type, model, speed. | SELECT DISTINCT Product.type, Laptop.model, Laptop.speed  FROM Laptop  INNER JOIN Product ON Product.model = Laptop.model  WHERE speed < ALL (SELECT speed FROM PC) |

|  |  |
| --- | --- |
| **Exercise: 18** | **My solution:** |
| Find the makers of the cheapest color printers.  Result set: maker, price. | SELECT DISTINCT Product.maker, Printer.price  FROM Product, Printer  WHERE Printer.color = 'y' AND  Product.model = Printer.model AND  Printer.price = (SELECT MIN(price)  FROM Printer  WHERE color = 'y') |

|  |  |
| --- | --- |
| **Exercise: 19** | **My solution:** |
| 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 DISTINCT Product.maker, AVG(Laptop.screen) AS avg\_screen  FROM Laptop  JOIN Product ON Product.model = Laptop.model  GROUP BY Product.maker |

|  |  |
| --- | --- |
| **Exercise: 20** | **My solution:** |
| Find the makers producing at least three distinct models of PCs.  Result set: maker, number of PC models. | SELECT DISTINCT maker, COUNT(model) AS Count\_model  FROM Product  WHERE type = 'PC'  GROUP BY maker  HAVING COUNT(model) >= 3 |

|  |  |
| --- | --- |
| **Exercise: 21** | **My solution:** |
| Find out the maximum PC price for each maker having models in the PC table.  Result set: maker, maximum price. | SELECT DISTINCT Product.maker, MAX(PC.price) AS MAX\_price  FROM PC  JOIN Product ON Product.model = PC.model  GROUP BY Product.maker |

|  |  |
| --- | --- |
| **Exercise: 22** | **My solution:** |
| 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 DISTINCT speed, AVG(price)  FROM PC  WHERE speed > 600  GROUP BY speed |

|  |  |
| --- | --- |
| **Exercise: 23** | **My solution:** |
| 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 Product.type = 'PC' AND  PC.speed >= 750  INTERSECT  SELECT maker  FROM Product  JOIN Laptop ON Product.model = Laptop.model  WHERE type = 'Laptop' AND  Laptop.speed >= 750 |

|  |  |
| --- | --- |
| **Exercise: 24** | **My solution:** |
| List the models of any type having the highest price of all products present in the database. | WITH fin\_t AS (SELECT model, price  FROM PC  UNION  SELECT model, price  FROM Laptop  UNION  SELECT model, price  FROM Printer)  SELECT model  FROM fin\_t  WHERE price = (SELECT MAX(price) FROM fin\_t) |

|  |  |
| --- | --- |
| **Exercise: 25** | **My solution:** |
| Find the printer makers also producing PCs with the lowest RAM capacity and the highest processor speed of all PCs having the lowest RAM capacity.  Result set: maker. | SELECT DISTINCT maker  FROM Product  WHERE model IN (  SELECT model  FROM PC  WHERE ram = (SELECT MIN(ram)  FROM PC) AND  speed = (SELECT MAX(speed)  FROM PC  WHERE ram = (SELECT MIN(ram)  FROM PC)  )  )  AND  maker IN (SELECT maker  FROM Product  WHERE type='printer'  ) |

|  |  |
| --- | --- |
| **Exercise: 26** | **My solution:** |
| Find out the average price of PCs and laptops produced by maker A.  Result set: one overall average price for all items. | WITH fin\_t AS (SELECT PC.price, PC.model, Product.maker, Product.type  FROM PC  JOIN Product ON Product.model = PC.model  WHERE Product.maker = 'A'  UNION ALL  SELECT Laptop.price, Laptop.model, Product.maker, Product.type  FROM Laptop  JOIN Product ON Product.model = Laptop.model  WHERE Product.maker = 'A')  SELECT AVG(price) AS AVG\_price  FROM fin\_t |

|  |  |
| --- | --- |
| **Exercise: 27** | **My solution:** |
| Find out the average hard disk drive capacity of PCs produced by makers who also manufacture printers.  Result set: maker, average HDD capacity. | SELECT DISTINCT Product.maker, AVG(hd) AS Avg\_hd  FROM PC  JOIN Product ON Product.model = PC.model  WHERE Product.maker IN (SELECT maker  FROM product  WHERE type='printer')  GROUP BY Product.maker |

|  |  |
| --- | --- |
| **Exercise: 28** | **My solution:** |
| Using Product table, find out the number of makers who produce only one model. | WITH final\_t AS (SELECT COUNT(maker) as qty  FROM Product  GROUP BY maker)  SELECT COUNT(qty) AS qnty  FROM final\_t  WHERE qty = 1 |

**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.

|  |  |
| --- | --- |
| **Exercise: 29** | **My solution:** |
| Under the assumption that receipts of money (inc) and payouts (out) are registered not more than once a day for each collection point [i.e. the primary key consists of (point, date)], write a query displaying cash flow data (point, date, income, expense).  Use Income\_o and Outcome\_o tables. | SELECT DISTINCT Outcome\_o.point, Outcome\_o.date, Income\_o.inc, Outcome\_o.out  FROM Outcome\_o  LEFT JOIN Income\_o ON Income\_o.date = Outcome\_o.date AND Income\_o.point = Outcome\_o.point  UNION  SELECT DISTINCT Income\_o.point, Income\_o.date, Income\_o.inc, Outcome\_o.out  FROM Income\_o  LEFT JOIN Outcome\_o ON Income\_o.date = Outcome\_o.date AND Income\_o.point = Outcome\_o.point |

|  |  |
| --- | --- |
| **Exercise: 30** | **My solution:** |
| Under the assumption that receipts of money (inc) and payouts (out) can be registered any number of times a day for each collection point [i.e. the code column is the primary key], display a table with one corresponding row for each operating date of each collection point.  Result set: point, date, total payout per day (out), total money intake per day (inc).  Missing values are considered to be NULL. | WITH final\_t AS (SELECT Income.point, Income.date, Outcome.out AS outcome, Income.inc AS income  FROM Income  LEFT JOIN Outcome ON Income.code = Outcome.code AND Income.point = Outcome.point  AND Outcome.date = Income.date  UNION ALL  SELECT Outcome.point, Outcome.date, Outcome.out AS outcome, Income.inc AS income  FROM Outcome  LEFT JOIN Income ON Income.code = Outcome.code AND Income.point = Outcome.point  AND Outcome.date = Income.date)  SELECT point, date, SUM(outcome) AS outcome, SUM(income) AS income  FROM final\_t  GROUP BY final\_t.point, final\_t.date |