Prof. Dr. Gunter Saake
Department of Technical and Business Information Systems
Workgroup Databases & Software Engineering

Database Concepts

Exercise 8

Given following relational schema:

$flight$	$(\underline{label, date}, destination, fligth\_time, distance)$
$departure$	$\Big(\underline{(flight\_label, flight\_date) \rightarrow flight, (type, serial\_number) \rightarrow plane}, captain)\Big)$
$employee$	$(\underline{ID}, name, address, job, salary)$
$plane\_type$	$(\underline{type}, manufacturer, number\_of\_seats, cruising\_speed)$
$plane$	$(\underline{type \rightarrow plane\_type, serial\_number}, purchase\_date, flight\_hours)$
$spare\_part$	$(\underline{ID}, label, price)$
$requires$	$(\underline{type \rightarrow plane\_type, serial\_number \rightarrow plane, spare\_part\_ID \rightarrow spare\_part})$
$pilot$	$(\underline{employee\_ID \rightarrow employee, license}, flight\_hours)$
$technician$	$(\underline{employee\_ID \rightarrow employee, team\_number})$
can_fly	$(\underline{employee\_ID \rightarrow pilot, type \rightarrow plane\_type})$
$can\_maintain$	$(\underline{employee\_ID \rightarrow technician, type \rightarrow plane\_type})$
$passenger$	$(\underline{passenger\_ID}, name, address, age)$
$booking$	$\Big(\underline{passenger\_ID \rightarrow passenger, (flight\_label, flight\_date) \rightarrow flight},$ $class, seat\_number, price)$

1. Formulate following queries in SQL:

   (a) Query name and salary of all employees that earn more than 6450€ .

   (b) Query name and salary of all employees that do not earn between 6000€ and 10000€ .

   (c) Query all planes that are of type $A-340$ or $TRIDENT$. Sort by purchase date.

   (d) Which pilots have a license different from I and II

   (e) Determine the names of employees that have an A at the third position of their name.

   (f) Determine the names of employees that have the L twice in their name.

   (g) Query name, job and salary of all employees whose job is either $Dipl.-Ing.$ or $steward/-ess$ and that earn are least 6000€ .

2. Define the following queries in SQL:

   (a) Determine employee_Id, name and salary of all employees. Add an intermediate column *new_salary* to the query result that shows the current salary increased by 15%. The *new_salary* must be returned as integer value.

   (b) Given your solution from **task (a)**, add another intermediate column that shows the difference between the original *salary* and the *new_salary*. The difference must also be returned as integer value.

   (c) For every plane, list its type, serial number and operating hours. Operating hours must be returned in a column called *operating_hours*.
   The operating hours are computed by calculating the difference between today and the purchase date of the respective plane. Use a standard date function to calculate the difference and return your result as integer. Finally, sort your list by *operating_hours*.

   (d) Create a query that returns following string for every employee:
   $< name >$ earns $< salary >$ per month, but desires to earn $< 3{*}salary >$.
   Replace all placeholders with the respective data using SQL. The new column is called *desired_salary*.

   (e) List all plane types. Thereby, all first letters must be capitalized, the rest must be uncapitalized. Return the length of the type name in a second column. The columns are called *name* und *length*.

3. Use SQL to retrieve following information:

   (a) How many planes (not types) are stored in relation *departure*?

   (b) Determine the number of employees that have a doctor's degree (*Dr.* or *PhD*)!

   (c) What is the *average_salary* by job?

   (d) Return the *total_price* and the *number_of_bookings* for all journeys in 1993 in relation *booking*.

   (e) Determine the minimum salary for every job!

   (f) Retrieve the *difference* between the maximum and minimum salary of the all employees.

4. You are new in the university IT department. Your task is to reformulate following SQL query as it is to slow currently:

```
SELECT   DISTINCT X.exam_ID FROM exams X
WHERE    X.exam_ID IN (
         SELECT Y.exam_ID FROM exams Y
         WHERE Y.student_ID <> X.student_ID);
```

   Maybe, a reformulation will improve the query performance. In order to reformulate it, solve following tasks:

   (a) What is the result of this query?

   (b) You found two ways to reformulate the query. Reformulate the query

      i. Without using a nested query in the WHERE clause!

      ii. Without using tuple variables (to distinguish the use of the same relation), but by using aggregation functions.

Good Luck!