ISSN: 2993-8759

Iris Journal of Educational Research

DOI: 10.33552/IJER.2023.01.000510

Iris Publishers

Short Communication

Evergreen Databases: A Practical Tool for Student's Training

Jozef Hvorecký*

Faculty of Education, University of Ostrava, Czech Republic

***Corresponding author:** Jozef Hvorecký, Faculty of Education, University of Ostrava, Czech Republic.

Received Date: August 13, 2023

Published Date: August 25, 2023

Introduction

To motivate students, the Database course should be full of real-life tasks. Their formulation should require the learners to concentrate on the currently trained topic, i.e., not to expect solving additional ones which should distract them. Among other limitations, it for example means that records containing dates must be authentic i.e., to refer to the period students know from their daily practice and are ready to accept them with no discussion on their relevance.

Let us discuss a query producing all records from a travel agency database responding to the client request: Show me all trips I can take part in in the next four weeks. Table 1 shows its first ten records (Table 1).

Table 1: Ten records in the database Trips.

Trips							
ID	**Destination**	**DateOfTrip**	**Transport**	**Price**	**MaxParticipants**	**Guide**	**Contact**
1	Modra	Tuesday, 24 October 2023	Bicycle	37 €	15	Miki	0903 231 387
2	Zohor	Sunday, 3 September 2023	Train	18 €	50	Lea	0244 600 872
3	Pezinok	Saturday, 19 August 2023	Bus	29 €	40	Tibor	0907 699 236
4	Malacky	Tuesday, 21 November 2023	Bicycle	12 €	22	Miki	0903 231 387
5	Modra	Tuesday, 2 January 2024	Bus	40 €	42	Andrej	0259 806 402
6	Trnava	Friday, 29 September 2023	Train	49 €	30	Mária	0910 774 916
7	Stupava	Friday, 8 September 2023	Bicycle	17 €	24	Andrej	0259 806 402
8	Skalica	Thursday, 24 August 2023	Train	29 €	38	Tibor	0907 699 236
9	Šaštín	Friday, 22 September 2023	Bus	39 €	42	Miki	0903 231 387
10	Zohor	Saturday, 18 November 2023	Train	22 €	30	Lea	0244 600 872

First, the database of trips must be prepared in advance and the students should not build it. Its creation would take a long time which could be devoted to other queries i.e., to deeper knowledge of the subject. Even though the table is short and contains a few records, its design and its value input would take dozens of minutes. Accessing the ready-made table saves this time.

The query solving the client's request sounds:

```
SELECT *
FROM Trips
WHERE DateOfTrip <= Date() + 28
```

Let us presume that the table was formed on 13 August 2023. On this date, four records will be extracted – see Table 2. As time will go, the query will produce different results. For example, after 24 August, it will show just three of them. Finally, after 2nd January 2024, an empty data set will be the response. The situation is realistic, and the students understand why it happens (Table 2).

Table 2: Records selected by query.

Available Trips							
ID	**Destination**	**DateOfTrip**	**Transport**	**Price**	**MaxParticipants**	**Guide**	**Contact**
2	Zohor	Sunday, 3 September 2023	Train	18 €	50	Lea	0244 600 872
3	Pezinok	Saturday, 19 August 2023	Bus	29 €	40	Tibor	0907 699 236
7	Stupava	Friday, 8 September 2023	Bicycle	17 €	24	Andrej	0259 806 402
8	Skalica	Thursday, 24 August 2023	Train	29 €	38	Tibor	0907 699 236

On the other hand, the lecturers are in a less convenient position. They must modify the table every time they plan to exploit such permanent records in their course. The database gets older: in one day non-updated tables will produce empty sets only.

Manual editing is tidy and may produce errors. This discomfort can be avoided by creating "evergreen databases". In such databases, some attributes are always updated when it is open. The same query will always generate the same number of records. In our case, the database will contain the replica of the table at Table 1, but its third column contains integers – the difference between "today" (the date of modification) and the desired value of DateOfTrip attribute.

In addition to the data for students, the evergreen database contains more tables - two in our case. The records for students will be in table Trips as before. From the table named DataSource, the educator will generate them. During the opening the records in the Trips table are delete using the action query

DELETE *

FROM Trips

Then new records to added to it using:

INSERT INTO Trips (ID, Destination, DateOfTrip, Transport, Price, MaxParticipants, Guide, Contact)

SELECT ID, Destination, Date()+TimeShift AS DateOfTrip, Transport, Price, MaxParticipants, Guide, Contact

FROM DataSource;

To speed up the process, the queries can be included into the Autoexec macro. Such an approach guarantees that there is always a database with fresh data and production at the same time when used in education. The author and his colleagues have used this approach for a long time. Using this approach, all databases in the textbook [1] and in the problem solver [2] were modified using the described approach. Then, they were attached to the books on a memory key. The readers often express their satisfaction with the approach.

Acknowledgement

None.

Conflict of Interest

No conflict of interest.

References

1. J Hvorecký (2013) Database technologies. Equilibria, Košice, p. 316.
2. J Hvorecký (2013) Database technologies: Supporting teaching material. Equilibria, Košice, p. 106.