M. S. Unanian

School of Business of BSU, Minsk, Belarus
Scientific supervisor – E.G. Grinevich, Senior lecturer

INFORMATION SYSTEM FOR AN INTERNATIONAL STUDENT DORMITORY

This scientific work focuses on creating a unified dormitory management system to address inefficiencies in manual and fragmented data processes. The system integrates room allocation, student records, facility booking, and event planning into a centralized database, streamlining operations and minimizing errors. The project highlights the benefits of modern technologies in improving efficiency, scalability, and service quality for managing international student accommodations.

Keywords: *database development, MS SQL server, table creation, query creation, dormitory management, information technology*

The research object of this study is an information system for managing activities related to university dormitories, including the registration of student accommodations and the administration of dormitory facilities. The focus is on developing a comprehensive database and information system to optimize dormitory management processes.

The purpose of this research is to design and implement a robust database and information system tailored for university dormitory administration. This system aims to enhance operational efficiency, streamline complex processes, and ensure secure, reliable data management for both dormitory staff and students.

This study also addresses limitations within current dormitory management systems, such as inefficiencies in data handling, the lack of integrated systems for managing student accommodations, and the challenge of maintaining secure and organized records. By providing a solution that simplifies these processes, the study aims to improve the overall management of university dormitories.

The user group includes the dormitory staff responsible for the overall management of the dormitory facilities, student accommodations, maintenance, and excursion planning.

Information Needs of Dormitory Administrators:

– Room Allocation and Management

Access to comprehensive details about each room, such as occupancy status, room condition, and amenities available.

Ability to assign rooms to students and manage changes based on student preferences or maintenance needs.

– Student Management

Detailed profiles of students, including personal information, contact details, and accommodation history.

Tools to monitor and manage student check-ins and check-outs, including tracking the duration of each stay.

– Facility Booking Management

System to manage and oversee the booking of facilities within the dormitory, such as study rooms, lounges, or recreational areas.

Ability to view booking schedules, adjust availability, and manage pricing for facility use.

– Excursion Planning and Management

Tools to organize and manage excursions, including details on excursion destinations, transportation, pricing, and student participation.

Ability to create, modify, and cancel excursions, as well as manage student sign-ups and payments.

Entities Description

The database consists of several key entities, each with specific attributes to capture and facilitate aspects of student accommodations and activities. Below, each entity is explored in detail.

1. Student

Attributes: StudentID (Primary Key), Name, Surname, Birthday, Nationality, Phone

2. Room

Attributes: RoomID (Primary Key), FloorID (Foreign Key), Number, Section, Conditioner Floor entity.

3. Floor

Attributes: FloorID (Primary Key), ManagerID (Foreign Key), Conditioner, Facilities

4. Manager

Attributes: ManagerID (Primary Key), ManagerName, Contacts

5. Excursion

Attributes: ExcursionID (Primary Key), Destination, Quantity, Transport, Time.

6. Booking

Attributes: BookingID (Primary Key), StudentID (Foreign Key), Facility, PricePerHour, Date.

7. Student_Excursion

Attributes: StudentID (Foreign Key), ExcursionID (Foreign Key), Date, Price.

8. CheckIn_Out

Attributes: CheckID (Primary Key), RoomID (Foreign Key), StudentID (Foreign Key), Date_Start, Date_End

9. SignOut

Attributes: SignOutID (Primary Key), StudentID (Foreign Key), Destination, Date_Start, Date_End

Each entity is linked through primary and foreign keys, establishing a strong relational structure that enables complex queries and extensive reporting capabilities. This network allows for tracking student accommodation history, managing excursion details, and overseeing facility bookings efficiently.

Draw the first version of the conceptual schema in any apps. The best app is ERWin Data Modeler (Fig.1).

Then we focus on implementing the dormitory database using SQL Server Management Studio, a free and user-friendly platform for database management.

The database, named Solbridge_dormitory, is initialized by creating tables using SQL scripts.

Indexing is a powerful tool in SQL databases, essential for optimizing query performance and ensuring efficient data retrieval. It transforms cumbersome searches into swift operations, ensuring that our dormitory management system runs smoothly and efficiently [4].

When populating a database, it's important to start with tables that do not depend on other tables for their data. This typically means beginning with tables that serve as references or look-up tables, often containing static or foundational data [5].

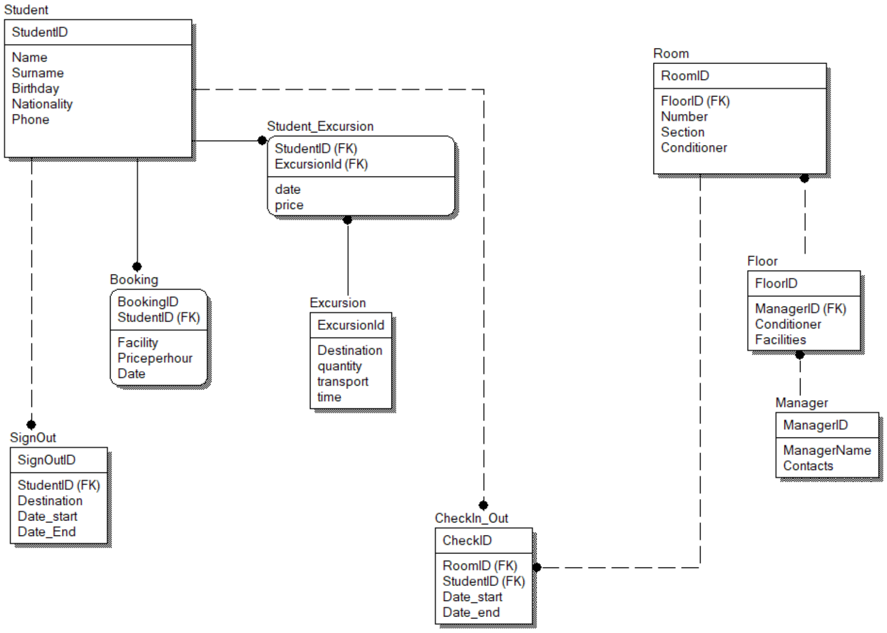
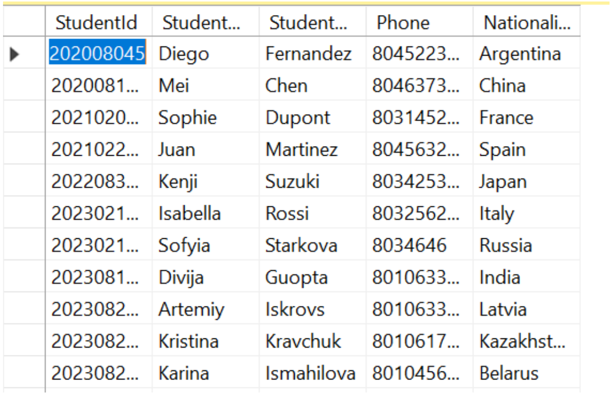


Fig. 1. Data Modelling

The process starts with Manager, followed by Floor, Room, and Students. An example of a populated table is shown in Fig. 2, highlighting the structured approach to data entry.

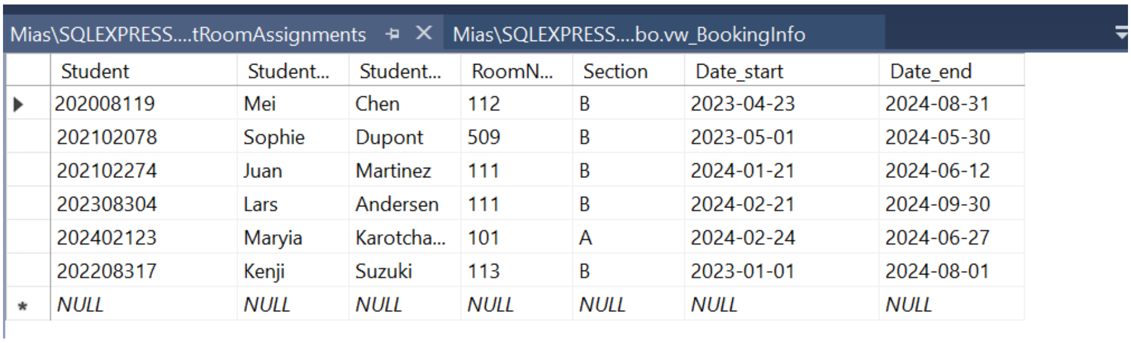


	StudentId	Student...	Student...	Phone	Nationali...
▶	202008045	Diego	Fernandez	8045223...	Argentina
	2020081...	Mei	Chen	8046373...	China
	2021020...	Sophie	Dupont	8031452...	France
	2021022...	Juan	Martinez	8045632...	Spain
	2022083...	Kenji	Suzuki	8034253...	Japan
	2023021...	Isabella	Rossi	8032562...	Italy
	2023021...	Sofyia	Starkova	8034646	Russia
	2023081...	Divija	Guopta	8010633...	India
	2023082...	Artemiy	Iskrovs	8010633...	Latvia
	2023082...	Kristina	Kravchuk	8010617...	Kazakhst...
	2023082...	Karina	Ismahilova	8010456...	Belarus

Fig. 2. Filled Students Table

After we have all the data in the tables we can start working on the views, functions, procedures and triggers.

Views are essential in a database for several reasons. They simplify complex queries by encapsulating them into a single, reusable query, which makes it easier for users to retrieve and understand data. Views provide a layer of abstraction over the actual database schema, allowing users to interact with a simplified representation of the data without needing to know the details of the underlying tables and relationships. Additionally, views can be used to restrict access to specific data, enhancing security. By granting permissions on views instead of tables, you can ensure that users only see and interact with the data they are authorized to access. Views also ensure that complex business logic and calculations are consistent across the application. Instead of duplicating logic in multiple queries, you define it once in a view and reuse it wherever needed. Lastly, in some cases, views can be optimized by the database engine to improve query performance, providing significant benefits for frequently accessed data [6]. An example can be found on the Fig. 3.


Mias\SQLEXPRESS....tRoomAssignments
Mias\SQLEXPRESS....bo.vw_BookingInfo

	Student	Student...	Student...	RoomN...	Section	Date_start	Date_end
▶	202008119	Mei	Chen	112	B	2023-04-23	2024-08-31
	202102078	Sophie	Dupont	509	B	2023-05-01	2024-05-30
	202102274	Juan	Martinez	111	B	2024-01-21	2024-06-12
	202308304	Lars	Andersen	111	B	2024-02-21	2024-09-30
	202402123	Maryia	Karotcha...	101	A	2024-02-24	2024-06-27
	202208317	Kenji	Suzuki	113	B	2023-01-01	2024-08-01
*	*NULL*	*NULL*	*NULL*	*NULL*	*NULL*	*NULL*	*NULL*

Fig. 3. View current Room Assignments

A stored procedure is a precompiled collection of one or more SQL statements stored in the database. It can be executed by calling it with specific parameters (if any), and it can perform various database operations such as querying, updating, inserting, or deleting data. Stored procedures are used to encapsulate repetitive tasks, ensure data integrity, and improve performance by reducing the amount of data sent between the client and the server [3].

User-defined functions (UDFs) in SQL Server are routines that accept parameters, perform specific tasks, and return a value. They are used to encapsulate logic for reuse, improve code readability, and perform operations that can be executed within SQL statements. There are two main types of UDFs: scalar functions and table-valued functions [2].

A trigger is a special kind of stored procedure that automatically executes or "fires" when specific database events occur. These events can be INSERT, UPDATE, or DELETE operations on a table or view. Triggers are often used to enforce business rules, data integrity, and audit changes to the data [1].

This trigger will decrement the "Quantity" column in the "Excursion1" table whenever a new row is inserted into the "ExcursionReg" table.

```
CREATE TRIGGER trg_ExcursionReg_AfterInsert
ON ExcursionReg
AFTER INSERT
AS
BEGIN
   DECLARE @ExcursionID INT;
   SELECT @ExcursionID = Excursion
   FROM inserted;
   UPDATE Excursion1
   SET Quantity = Quantity - 1
   WHERE ExcursionID = @ExcursionID AND Quantity > 0;
END;
```

In conclusion, the use of SQL queries, views, stored procedures, user-defined functions, and triggers significantly enhances the efficiency and management of the dormitory system. These tools streamline data processing, improve performance, ensure data integrity, and simplify repetitive tasks. By leveraging these SQL techniques, the system can effectively manage excursions, student information, and room assignments, ultimately supporting better decision-making and operational efficiency in the dormitory management process.

References

1. *Ben-Gan, I.* T-SQL Fundamentals / I. Ben-Gan. – Redmond : Microsoft Press, 2016. – 432 p.
2. Learn SQL // Codecademy. – URL: https://www.codecademy.com/learn/learn-sql (date of access: 01.11.2024).
3. *Goulet, D.* Learning SQL / D. Goulet, C. Sheridan. – Sebastopol : O'Reilly Media, 2020. – 384 p.
4. *Portilla, J.* The Complete SQL Bootcamp 2023: Go from Zero to Hero / J. Portilla // Udemy. – URL: https://www.udemy.com/course/the-complete-sql-bootcamp/ (date of access: 01.11.2024).
5. SQL для науки о данных // Coursera. – URL: https://www.coursera.org/learn/sql-for-data-science (дата обращения: 01.11.2024).
6. SQL Tutorial // W3Schools. – URL: https://www.w3schools.com/sql (date of access: 30.10.2024).