Week 2 Tutorial: Data Modelling for a Smart School Database System

Name: Emad Muhith

ID: 2516011

Date: 14th August 2026

Table of Contents

Introduction.....3

Part 1: Introduction to Data Modelling.....3

Part 2: Conceptual Data Modelling.....3

Part 3: Logical Data Modelling.....6

Part 4: Physical Data Modelling.....7

Task 2: Data Modelling for Assessment 1 Scenario.....7

Conclusion.....8

References.....9

Introduction

Data modelling is a systematic procedure that is employed to determine the manner in which data is organised, stored and connected in an information system. Data modelling is significant in system development as it assists developers and users to comprehend the data requirement prior to database implementation. A definite data model eliminates duplication, helps with data accuracy and offers a basis to sound database design [1], [2]. The Smart School Database System needs to store the classes, subjects and exam results, students, and teachers in a structured manner, which is achieved through data modelling.

Part 1: Introduction to Data Modelling

The term data modelling can be defined as the process of recognizing data entities, their attributes and describing the relationships between them. It serves as a blue print to design database and facilitates interaction between users, analysts and developers [2]. Badly designed data models may pose challenges in relationships, attributes and cardinalities, which can translate to the quality of the end database [4].

Data modelling is of three major types. The former is conceptual data modelling, which gives the high-level perspective of the key entities and the relationships. The second one is logical data modelling which provides attributes, primary keys, foreign keys and relation rules. The third one is physical data modelling that transforms the logical model into database tables, fields, data types and constraints to be implemented in a particular database system [1], [5].

Part 2: Conceptual Data Modelling

The initial step of database design is conceptual data modelling. It dwells on the central business entities and their relationships between each other. No technical information like the type of the fields and software settings is provided at this stage. This is to reflect what data the organisation should be dealing with [5].

In the case of the Smart School Database System, Student, Teacher, Class, Subject and ExamResult are the primary entities. A student is a part of a class and a class is a part of a student. A teacher may teach either one or more subjects and each subject may be taught by a teacher. A

student is allowed to do examinations in various subjects and the outcome of the exams is a record of marks and grades in a particular subject. Studies on database learning indicate that students usually go wrong when defining relationships and cardinalities and hence these connections should be well defined [4], [6].

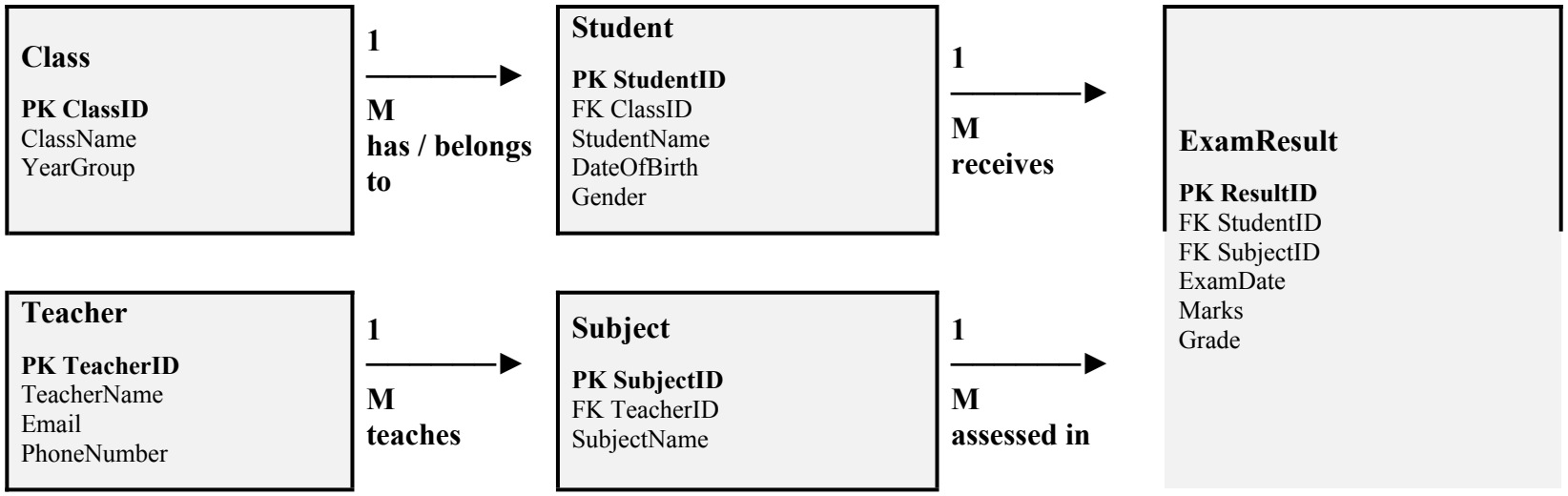

Figure 1: Conceptual ERD for Smart School Database System

Relationship summary

Parent entity	Child entity	Relationship	Cardinality
Class	Student	One class has many students	1:M
Teacher	Subject	One teacher teaches many subjects	1:M
Student	ExamResult	One student receives many exam results	1:M
Subject	ExamResult	One subject appears in many exam results	1:M

PK = Primary Key | FK = Foreign Key | 1:M = One-to-Many Relationship

ExamResult entity is significant as it links students and subjects. This prevents marks being stored directly in the Student or Subject table and makes it easier to get more accurate academic reporting.

Part 3: Logical Data Modelling

Conceptual to logical data modelling consists in making it more detailed. Attributes are assigned to entities at this stage, primary keys are determined and foreign keys are also added to indicate relationships among the tables [2]. Logical modelling is independent of a platform, i.e. it can be written before deciding on the actual database software.

Entity / Table	Primary Key	Foreign Key(s)	Main Attributes	Relationship Role
Class	ClassID	None	ClassName, YearGroup	One class has many students
Student	StudentID	ClassID	StudentName, DateOfBirth, Gender	Each student belongs to one class and can have many exam results
Teacher	TeacherID	None	TeacherName, Email, PhoneNumber	One teacher can teach many subjects
Subject	SubjectID	TeacherID	SubjectName	Each subject is taught by one teacher and can appear in many results
ExamResult	ResultID	StudentID, SubjectID	ExamDate, Marks, Grade	Links each student to a subject result

In this model, StudentID, TeacherID, ClassID, SubjectID and ResultID are primary keys. ClassID, TeacherID, and StudentID and SubjectID are foreign keys in Student, Subject, and ExamResult, respectively. These keys are used to preserve referential integrity, i.e. records are properly linked [1], [2] together. Proper labeling of entities, attributes and relationships is also significant as ambiguous labels can decrease the usefulness of the data models [4].

This logical model facilitates the running of schools since it adds the students that are enrolled in which classes, the teachers who teach which subject and the outcomes that each student has achieved.

Part 4: Physical Data Modelling

Physical data modelling transforms logical data model into the real database structures. These involve characterizing tables, fields, data types, primary keys, foreign keys and validation rules [2]. Understandably, StudentID can be AutoNumber in Microsoft Access, StudentName can be Short Text, DateOfBirth can be Date/Time and Marks can be Number.

Table	**Primary Key**	**Foreign Key**
Class	ClassID	None
Student	StudentID	ClassID
Teacher	TeacherID	None
Subject	SubjectID	TeacherID
ExamResult	ResultID	StudentID, SubjectID

The relationships are implemented by linking primary keys to foreign keys. For example, Class.Student is connected to ClassID.ClassID, Teacher.Subject is connected to TeacherID.TeacherID, Student.ExamResult is connected with StudentID.StudentID and Subject.ExamResult is related to SubjectID.SubjectID. These links assist in making sure that a result cannot be typed in on a student or subject that is not available.

In physical database design, normalisation is also a matter of concern. It normalises data to minimise the redundancy and prevent update, insertion and deletion anomalies [1], [3]. As an instance, the Teacher table should store all the teacher information, rather than duplicating it in each Subject record. Recent applied database research also demonstrates the usefulness of structured relational schema to store and retrieve data effectively [7], [8]. The reverse is denormalisation which entails combining or duplicating some data to enhance the speed of reporting. Nevertheless, redundancy may be enhanced through denormalisation and must have a rationale in performance requirements [9], [10].

Task 2: Data Modelling for Assessment 1 Scenario

In Assessment 1, the scenario is the AutoPark Management System. The conceptual, logical and physical steps of data modelling in this case should be the same. The key concepts would be ParkingSite, ParkingSlot, Customer, Vehicle, Booking and Payment on the conceptual stage. These entities are the fundamental information needed to run parking operations.

Attributes and keys would be added at the logical stage. ParkingSite would have SiteID, SiteName and Address. ParkingSlot would consist of SlotID, SiteID, Slotnumber and Status. Customer would encompass CustomerID, FullName and PhoneNumber. Vehicle would include VehicleID, CustomerID and RegistrationNumber. Booking would consist of BookingID, CustomerID,

VehicleID, SlotID, EntryTime and ExitTime. It would be paid in the form of PaymentID, BookingID, AmountPaid and PaymentStatus.

At the physical level, these entities would be represented in form of tables in Microsoft Access. Records would be uniquely identified by primary keys, whereas foreign keys would have relationships between related tables. For example, ParkingSite.SiteID would be linked to ParkingSlot.SiteID, Customer.Vehicle would be linked to CustomerID.CustomerID and Booking.CustomerID, and Booking.Payment would be linked with BookingID.BookingID. This is in line with principles of relational database, wherein structured relationships enhance data integrity and reliability in querying [1], [2]. The NoSQL databases can be convenient in certain large scale or flexible data environments but the relational database is still applicable where there are structured records, relationship and transactional consistency is needed [11], [12].

Conclusion

The stage of data modelling is required during the development of a database system as it gives a clear format of how to store and manipulate organisational data. The Smart School Database System illustrates how a general business requirement can be converted to a specific database design by way of conceptual, logical and physical models. Conceptual modelling determines the primary entities, logical modelling adds keys and attributes and physical modelling readies the system to be implemented. This process may as well be used on the AutoPark Assessment 1 scenario to generate a trustworthy, relational and normalised database.

References

[1] R. Elmasri and S. B. Navathe, *Fundamentals of Database Systems*, 7th ed. Harlow, UK: Pearson, 2016.

[2] T. M. Connolly and C. E. Begg, *Database Systems: A Practical Approach to Design, Implementation, and Management*, 6th ed. Harlow, UK: Pearson, 2015.

[3] C. J. Date, *An Introduction to Database Systems*, 8th ed. Boston, MA: Addison-Wesley, 2004.

[4] K. Rosenthal, S. Strecker and M. Snoeck, "Modeling difficulties in creating conceptual data models: Multimodal studies on individual modeling processes," *Software and Systems Modeling*, vol. 22, pp. 1005–1030, 2023, doi: 10.1007/s10270-022-01051-8.

[5] R. Lukyanenko, B. M. Samuel, J. Parsons, V. C. Storey, O. Pastor and A. Jabbari, "Universal conceptual modeling: Principles, benefits, and an agenda for conceptual modeling research," *Software and Systems Modeling*, vol. 23, no. 5, pp. 1077–1100, 2024, doi: 10.1007/s10270-024-01207-8.

[6] N. Abdelaziz, M. Farghally, M. Mohammed and T. H. A. Soliman, "Identifying and addressing common undergraduate database design misconceptions," *Frontiers in Education*, vol. 10, 2025, doi: 10.3389/feduc.2025.1554089.

[7] D. Trillo-Montero, S. Cosano-Lucena, M. Gonzalez-Redondo, J. J. Luna-Rodriguez and I. Santiago, "Design and development of a relational database management system (RDBMS) with open source tools for the processing of data monitored in a set of photovoltaic (PV) plants," *Applied Sciences*, vol. 13, no. 3, article 1357, 2023, doi: 10.3390/app13031357.

[8] I. S. Panagea, A. Dangol, M. Olijslagers, J. Diels and G. Wyseure, "A database schema for standardized data and metadata collection in agricultural experiments," *Land*, vol. 14, no. 9, article 1816, 2025, doi: 10.3390/land14091816.

[9] D. Martinez-Mosquera, R. Navarrete and S. Lujan-Mora, "Modeling and management Big Data in databases: A systematic literature review," *Sustainability*, vol. 12, no. 2, article 634, 2020, doi: 10.3390/su12020634.

[10] P. Kotiranta, M. Junkkari and J. Nummenmaa, "Performance of graph and relational databases in complex queries," *Applied Sciences*, vol. 12, no. 13, article 6490, 2022, doi: 10.3390/app12136490.

[11] V. F. de Oliveira, M. A. d. O. Pessoa, F. Junqueira and P. E. Miyagi, "SQL and NoSQL databases in the context of Industry 4.0," *Machines*, vol. 10, no. 1, article 20, 2022, doi: 10.3390/machines10010020.

[12] W. Khan, T. Kumar, C. Zhang, K. Raj, A. M. Roy and B. Luo, "SQL and NoSQL database software architecture performance analysis and assessments: A systematic literature review," *Big Data and Cognitive Computing*, vol. 7, no. 2, article 97, 2023, doi: 10.3390/bdcc7020097.