Assignment 05

Eternal University

Document Details

Submission ID
trn:oid:::11424:149826270

Submission Date
Aug 29, 2026, 12:20 PM GMT+5:30

Download Date
Aug 29, 2026, 1:39 PM GMT+5:30

File Name
Humanizer_File_34.docx

File Size
109.3 KB

6 Pages

1,074 Words

6,322 Characters

0% detected as AI

The percentage indicates the combined amount of likely AI-generated text as well as likely AI-generated text that was also likely revised by AI.

Caution: Review required.

It is essential to understand the limitations of AI detection before making decisions about a student's work. We encourage you to learn more about Turnitin's AI detection capabilities before using the tool.

0 AI-generated 0%

Likely generated or likely generated and revised by AI.

Disclaimer

Our AI writing assessment is designed to help educators identify text that might be prepared by a generative AI tool. Our AI writing assessment may not always be accurate (it may misidentify writing that is likely human generated as AI generated and likely AI generated as human generated) so it should not be used as the sole basis for adverse actions against a student. It takes further scrutiny and human judgment in conjunction with an organization's application of its specific academic policies to determine whether any academic misconduct has occurred.

DATABASE DESIGN AND MANAGEMENT

turnitin Page 3 of 8 - AI Writing Submission Submission ID trn:oid:::11424:149826270

Table of Contents

1. Scenario and Purpose 3
2. Initial Unnormalised Design (UNF) 3
3. ChatGPT Prompt and Initial Response 4
4. Conclusion 5
References 5
Appendix A: ChatGPT Prompt and Response 6

turnitin Page 4 of 8 - AI Writing Submission Submission ID trn:oid:::11424:149826270

1. Scenario and Purpose

Database normalization has the aim to ensure that each piece of information is stored in the most appropriate place and minimize any unnecessary duplication. The primary objective is to ensure that each piece of information is stored in the most appropriate place and less unnecessary duplication is minimized. Normalization is generally performed through several normal forms, known as normal forms. First Normal Form is required each column to contain atomic values and eliminates repeating groups. Second Normal Form builds on the 1NF of 1NF by removing partial dependencies, ensuring that non-key attributes depend on the whole primary key. The Third Normal Form (3NF) further eliminates transitive dependencies, resulting in non-key attributes with a primary key or non-key attributes without other non-key attributes.

2. Initial Unnormalized Design (UNF)

The UNF is a general purpose document for defining the basic structure of a database, and is typically used to define the initial phase of development of the system. The design should be a single table of data, and be able to define all the data for the system before it is transformed into normalized tables.Represent the information that is stored in a system such as an airline booking system. In the example, a booking system with single PassengerID, PassengerName, Email, FlightID, FlightNumber, Destination, BookingID, SeatNumber, BookingID, BookingPlaces, and PaymentDetails can be adapted to include multiple ticket types. In the case of a complex ticket structure, you can change the information on your email by updating multiple records or deleting a booking. In the case of a recurring cancellation, there is one problem that you can have to find a new agent. If you delete a ticket, you will end up with more information, and thus, your ticket. First Normal Form (Fon) is a starting point for normalization of a database schema in which a single database schema is created. Second Normal Form is the best way of making a schema that is a logical unit for the database. Third Normal Form is a process for re-structuring of the database.

BookingID	PassengerID	Passenger Name	PassengerEmail	Passenger Phone	TicketIDs	FlightIDs	SeatNumbers	RouteDetails	AircraftDetails
B1001	P201	Aisha Khan	aisha@example.com	0400-110-200	T5001,T5002	QF401,QF401	12A,12B	SYD-MEL	A330-300 / 277 seats
B1002	P202	Daniel Lee	daniel@example.com	0400-110-201	T5003	QF412	14C	MEL-SYD	A330-300 / 277 seats
B1003	P203	Mia Wong	mia@example.com	0400-110-202	T5004,T5005	QF11,QF11	3A,3B	SYD-LAX	A380-800 / 484 seats
B1004	P204	Noah Smith	noah@example.com	0400-110-203	T5006	QF21	22D	SYD-DFW	A380-800 / 484 seats
B1005	P205	Olivia Chen	olivia@example.com	0400-110-204	T5007,T5008	QF7,QF7	8C,8D	SYD-DFW	787-9 / 236 seats
B1006	P206	Ethan Brown	ethan@example.com	0400-110-205	T5009	QF65	18A	BNE-SYD	737-800 / 174 seats
B1007	P207	Sofia Patel	sofia@example.com	0400-110-206	T5010,T5011	QF72,QF72	10A,10B	PER-SYD	737-800 / 174 seats
B1008	P208	Liam Jones	liam@example.com	0400-110-207	T5012	QF93	31F	MEL-LHR	787-9 / 236 seats

3. ChatGPT Prompt and Initial Response

MongoDB is a document-based data model. This is why the database is normalized differently than a relational database because it uses a document-based data model instead of tables and rows. Unlike the MongoDB database, these issues are normally resolved by separating the data in two distinct tables and creating tables of related data, but MongoDB allows related information to be stored in a single document using nested objects and arrays that can combine related information into one document. Query definitions in MongoDB are typically a human-readable structure, which is a good structure for normalized relational databases. The denormalization schema can be a good structure for the MongoDB schema, and it is possible to use this structure in MongoDB. However, excessive duplication can make updates and data consistency difficult. Relational databases are designed to help humans understand and represent data and document information in ways that can be easily manipulated, allowing the user to create a more humanly responsive user experience. MongoDB is

an open source database that provides a level of flexibility that makes it easier to create custom applications that focus on normalization and relationships between tables and allows developers to choose between embedding related data within documents and referencing information stored in separate collections.

4. Conclusion

Database normalization is an important process for creating a consistent, consistent, and efficient database structure. This process helps decrease duplication, remove unnecessary dependencies, prevent insertion, deletion, and update anomalies. MongoDB is a flexible document-based database that has multiple keys that allow it to store information in multiple databases, but which does not require multiple tables to be built and embedded within documents. The user interface is poorly designed and designed to maximize the performance of MongoDB – and it will continue to be so if data duplication isn't allowed to improve performance. Relational databases offer applications with many benefits in terms of data integrity and relationships, while MongoDB offers greater flexibility for applications containing changing, hierarchical, or document-oriented data.

References

Mukhlis, I.R., Hermansyah, D. and Lantang, V.M., 2023. Rancangan Basis Data Transaksi Pada PT. Bank Perkreditan Rakyat ABC Menggunakan MySQL Dengan Model Entity Relationship Diagram

(ERD) dan Physical Data Model (PDM). *Journal of Advances in Information and Industrial Technology*, *5*(1), pp.1-10.

Patel, S. and Choudhary, J., 2024, July. Metaphorical Conversion of Conceptual Model from Relational to NoSQL. In *International Conference on Data Science and Big Data Analysis* (pp. 277-286). Singapore: Springer Nature Singapore.

Taipalus, T., 2025. On the effects of logical database design on database size, query complexity, query performance, and energy consumption. *arXiv preprint arXiv:2501.07449*.

Tang, D., Zhao, R., Lin, Y., Zhang, T. and Zhang, P., 2022. Modeling the data provenance of relational databases supporting full-featured SQL and procedural languages. *Applied sciences*, *13*(1), p.64.

Trillo-Montero, D., Cosano-Lucena, S., Gonzalez-Redondo, M., Luna-Rodriguez, J.J. and Santiago, I., 2023. 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*, *13*(3), p.1357.

Copeland, D.G. and McKenney, J.L., 1988. Airline reservations systems: lessons from history. *MIS quarterly*, *12*(3), pp.353-370.

Beckmann, M.J., 1958. Decision and team problems in airline reservations. *Econometrica: journal of the Econometric Society*, pp.134-145.

Pokhrel, S., Ganesan, S., Banjade, S.R. and Somasiri, N., 2024. Enhancing the Usability, Visibility, and Responsiveness of an Airline Reservation System: A User-Centered Design Approach. *International journal of computer communication and informatics*, *6*(1), pp.1-15.

Appendix A: ChatGPT Prompt and Response

Possible Functional Dependencies

Based on the original table, the following functional dependencies may exist:

1. **BookingID → PassengerID, PassengerName, PassengerEmail, PassengerPhone, TicketIDs, FlightIDs, SeatNumbers, RouteDetails, AircraftDetails**
2. **PassengerID → PassengerName, PassengerEmail, PassengerPhone**
3. **TicketID → BookingID**
4. **TicketID → FlightID, SeatNumber**
5. **FlightID → RouteDetails, AircraftDetails**
6. **SeatNumber, FlightID → TicketID**
   (Assuming each seat can only be assigned to one ticket on a particular flight.)
7. **PassengerEmail → PassengerID**
   (Assuming each passenger has a unique email address.)
8. **PassengerPhone → PassengerID**
   (Assuming each passenger has a unique phone number.)

These are the likely functional dependencies for the original unnormalised booking table, without applying any normalisation.