

Assessor Feedback:**LO1 Use an appropriate design tool to design a relational database system for a substantial problem**Pass, Merit & Distinction P1 M1 D1
Descripts**LO2 Develop a fully functional relational database system, based on an existing system design**Pass, Merit & Distinction P2 P3 M2 M3 D2
Descripts**LO3 Test the system against user and system requirements.**Pass, Merit & Distinction P4 M4 D2
Descripts**LO4 Produce technical and user documentation.**Pass, Merit & Distinction P5 M5 D3
Descripts**Higher Nationals - Summative Assignment Feedback Form**

Student Name/ID			
Unit Title	Unit 04 – Database Design & Development		
Assignment Number	1	Assessor	
Submission Date		Date Received 1st submission	
Re-submission Date		Date Received 2nd submission	

* Please note that grade decisions are provisional. They are only confirmed once internal and external moderation has taken place and grades decisions have been agreed at the assessment board.

Assessor Feedback:

Grade:	Assessor Signature:	Date:
Resubmission Feedback: <ul style="list-style-type: none"> Please note resubmission feedback is focussed only on the resubmitted work 		
Grade:	Assessor Signature:	Date:
Internal Verifier's Comments:		
Signature & Date:		

- Please note that grade decisions are provisional. They are only confirmed once internal and board external moderation has taken place and grades decisions have been agreed at the assessment

BTEC HN Summative Assignment Feedback Form
 Issue Date: June 2021 Owner: HN QD
 DCL1 Public (Unclassified) Version 1.0

Important Points:

- It is strictly prohibited to use textboxes to add texts in the assignments, except for the compulsory information. eg: Figures, tables of comparison etc. Adding text boxes in the body except for the before mentioned compulsory information will result in rejection of your work.
- Avoid using page borders in your assignment body.
- Carefully check the hand in date and the instructions given in the assignment. Late submissions will not be accepted.
- Ensure that you give yourself enough time to complete the assignment by the due date.
- Excuses of any nature will not be accepted for failure to hand in the work on time.

6. You must take responsibility for managing your own time effectively.
7. If you are unable to hand in your assignment on time and have valid reasons such as illness, you may apply (in writing) for an extension.
8. Failure to achieve at least PASS criteria will result in a REFERRAL grade.
9. Non-submission of work without valid reasons will lead to an automatic RE FERRAL. You will then be asked to complete an alternative assignment.
10. If you use other people's work or ideas in your assignment, reference them properly using HARVARD referencing system to avoid plagiarism. You have to provide both in-text citation and a reference list.
11. If you are proven to be guilty of plagiarism or any academic misconduct, your grade could be reduced to A REFERRAL or at worst you could be expelled from the course
12. Use word processing application spell check and grammar check function to help editing your assignment.
13. **Use footer function in the word processor to insert Your Name, Subject, Assignment No, and Page Number on each page.** This is useful if individual sheets become detached for any reason.

STUDENT ASSESSMENT SUBMISSION AND DECLARATION

When submitting evidence for assessment, each student must sign a declaration confirming that the work is their own.

Student name:		Assessor name:
Issue date:	Submission date:	Submitted on:
Programme: Pearson BTEC HND in Computing		
Unit: Unit 04		
Assignment number and title: Database System for E-Space Solutions		

Plagiarism

Plagiarism is a particular form of cheating. Plagiarism must be avoided at all costs and students who break the rules, however innocently, may be penalised. It is your responsibility to ensure that you understand correct referencing practices. As a university level student, you are expected to use appropriate references throughout and keep carefully detailed notes of all your sources of

materials for material you have used in your work, including any material downloaded from the Internet. Please consult the relevant unit lecturer or your course tutor if you need any further advice.

Guidelines for incorporating AI-generated content into assignments:

The use of AI-generated tools to enhance intellectual development is permitted; nevertheless, submitted work must be original. It is not acceptable to pass off AI-generated work as your own.

Student Declaration

Student declaration

I certify that the assignment submission is entirely my own work and I fully understand the consequences of plagiarism. I understand that making a false declaration is a form of malpractice.

Student signature:

Date:

Unit 04: Database Design & Development

Assignment Brief

Student Name/ID Number	
Unit Number and Title	Unit 04 – Database Design & Development
Academic Year	2024/2025
Unit Tutor	
Assignment Title	Database System for E-Space Solution
Issue Date	
Submission Date	
Submission Format	

The assignment submission is in the form of the following.

- The submission should be in the form of an individual written report written in a concise, formal business style using single spacing and font size 12.
- **Completed functional relational database.** Final working version in a format suitable to be run and assessed for functionality – this could be as project/solution files or final compiled executable
- **A full Testing document**
- **Technical and User instructional videos** for successful use of the database. Use appropriate software and submit in a suitable format
- **written report** to evaluate the database and its implementation.

The recommended word limit is 3,000 – 3,500 words, although you will not be penalised for exceeding the total word limit.

All work must be supported with research and referenced using the Harvard referencing system

Unit Learning Outcomes

Use an appropriate design tool to design a relational database system for a
LO1 substantial problem

Develop a fully-functional relational database system, based on an existing
LO2 system design.

LO3 Test the system against user and system requirements.

LO4 Produce technical and user documentation.

Transferable skills and competencies developed

Computing-related cognitive skills :

- Demonstrate knowledge and understanding of essential facts, concepts, principles and theories relating to computing and computer applications
- Use such knowledge and understanding in the modelling and design of computer-based systems for the purposes of comprehension, communication, prediction and the understanding of trade-offs
- Recognise and analyse criteria and specifications appropriate to specific problems, and plan strategies for their solutions
- Critical evaluation and testing: analyse the extent to which a computer-based system meets the criteria defined for its current use and future development
- Methods and tools: deploy appropriate theory, practices and tools for the design, implementation and evaluation of computer-based systems.

Computing-related practical skills :

- The ability to specify, design and construct reliable, secure and usable computer-based systems
- The ability to evaluate systems in terms of quality attributes and possible trade-offs presented within the given problem
- The ability to deploy effectively the tools used for the construction and documentation of computer applications, with particular emphasis on understanding the whole process involved in the effective deployment of computers to solve practical problems
- The ability to critically evaluate and analyse complex problems, including those with incomplete information, and devise appropriate solutions, within the constraints of a budget.

Generic skills for employability:

- Intellectual skills: critical thinking; making a case; numeracy and literacy
- Self-management: self-awareness and reflection; goal setting and action planning
- Independence and adaptability; acting on initiative; innovation and creativity
- Contextual awareness, e.g. the ability to understand and meet the needs of individuals, business and the community, and to understand how workplaces and organisations are governed.

Vocational scenario

With marking a new chapter of human history **E-Space Solutions (Pvt.) Ltd** decided to colonize the planet Mars in 2040. The company decided to hire a database expert to develop a Database Management system to assist in this project for decision making. Therefore, the data should maintain accuracy & consistency throughout the data life cycle. E-Space Solutions plan to collect the information of the people who are willing to go to

Mars. To colonize Mars, the company needs to think about the qualifications of volunteers & how they can assist in the colonizing process on Mars. The people who willing to go to Mars will be known as '**Colonist.**' Each colonist will be identified with Mars Colonization ID, Name with First Name, Middle Name, Surname, Date of Birth, Qualification, Age, Earth Address, Gender, Contact No, Civil Status, number of members of the family to go to Mars & etc. From a single family there can be only one colonist & the colonist can decide who will go to the planet Mars with them. If desired, colonists can take their whole family with them

too. But they will be considered as Dependents of the colonist. In the system the dependents are not needed to identify separately, they should only need to identify according to the colonist. But the dependents details should be recorded with their Name, Date of Birth, Age, Gender, and Relationship to the colonist & etc.

E-Space Solutions planning to have develop space crafts which will be named E-Jets. Those E-Jets will take the colonists & their families together to the Mars & the pilots will return with empty E-Jet. Those E-Jet pilots will be well experienced astronomers who have a number of space hours. E-Jets will identify with Jet unique code, number of passenger seats, nuclear engine power, made year, weight, power source & etc. Company plan to build three types of E-Jets & they are,

- Jets which powered by only nuclear engine
- Jets which are powered by nuclear engine to go through the thick atmosphere close to the Earth surface & then use Hydro Splitter engine to go through the space with more than the speed of the sound & again it will use nuclear engines when landing on the Mars.
- Jets which are powered by Hydro-Nuc engine which use Nuclear power to split the water molecules & generate a super hydro power for the jet. But this technology will be high in cost, so at the initial level the company will have one of these only.

The astronomers who operate the E-Jets will have multiple qualifications in different levels. Also, they will have designations according to their ranks in the company. The pilots are assigned to the Jets rather than to the trips.

Those E-Jets will go multiple number of trips to Mars, with number of colonists & their families. Every colonist & family member (if they go with the colonist) will get a single trip to Mars & there's no return. According to the company the trip table should be as follows.

TripID	JetCode	LaunchDate	ReturnDate

The colonists will get a house from the Mars Colony. If the colonist going with the family, will get a single house or if not with the family the single house will assign to four colonists. The houses can be identified by unique Colony Lot Number, Number of Rooms, Square feet of the house & any other relevant record that needed.

Also, each & every colonist will be assigned a job in the colonization process. Some of the colonists will have multiple jobs. Those jobs can be Construction, Mechanical, Power Generation, Medical, Security, Administration, Education, Research & Observation & etc. One job will assign for multiple colonists.

Above details of the system will be input by the Data Entry Operators of the project. Several reports will be generated by the System administrator such as Jet Detail report, Trip Detail report with passengers, Colonists details with dependents & etc. Also, when the project is ongoing the Colony Superintendent will generate Colony report with House details & colonists, Job reports of the colonists & etc. The pilots can check the data of the system according to the space craft trip information with the passenger details.

E-Space solution is hiring you as the Database expert to design & develop a solution for the above scenario. In the System development level, the company doesn't need the report generated part initially, it will include in the next level of the development. But the Database designs & system designs should include the reporting processes.

Assignment activity and guidance

Based on E-Space Solutions scenario you will need to:

Activity 1 – User Requirements and Logical Design

- a. Identify the user and system requirements of the E-Space Solutions System.
- b. According to the above requirements that identified, design a database and design a relational database system using conceptual design (ER Model) with including identifiers (primary Key) of entities and attributes, cardinalities, participations of relationships.
- c. Convert the ER Model into logical database design using relational database model including primary keys foreign keys and referential Integrities. It should contain at least six interrelated tables.
- d. Check whether the provided logical design is normalized. If not, normalize the database by removing the anomalies.

Note: It is allowed to have your own appropriate assumptions and related attributes within the scope of the case study given

- e. Design set of simple interfaces to input and output for the above scenario using Wireframe or any interface-designing tool. (*Those designs cannot be the interface designs of visual studio.*)
- f. Evaluate the effectiveness of the given design (ERD and Logical design) in terms of the identified user and system requirements.

Activity 2 – Physical Design

- a. Develop a relational database system according to the ER diagram you have designed (**Use SQL DDL statements**). Should use data validation when develop the database.
- b. Provide evidence of the use of a suitable IDE to create a simple interface to insert, update and delete data in the database.
- c. Implement proper security mechanisms in the developed database. Evaluate the database solution developed and its effectiveness with relevant to the user and system requirements identified, system security mechanisms (EX: -User groups, access permissions) and the maintenance of the database.

*** Security mechanism should demonstrate with the necessary screen shots of the steps that follows during creating user groups, access permissions in the Microsoft SQL Sever.*

- d. Explain the usage of DML with below mentioned queries by giving at least two examples per each case from the developed database.

Select / Where / Insert / Update / Between / In / Group by / Order by / Having

- e. Assess the usage of the above SQL statements with the examples from the developed database to prove that the data extracted through them are meaningful & relevant to the given scenario.

Activity 3 -Testing

- a. Provide a suitable test plan to test the system against user & system requirements.

- b. Provide relevant test cases for the database you have implemented.
- c. Assess how the selected test data can be used to improve the effectiveness of testing.

Note: Learner need to give expected results in a tabular format & screenshots of the actual results with the conclusion.

Activity 4 – Review and Maintenance

- a. Get independent feedback on your database solution from the non-technical users and some developers (use surveys, questioners, interviews, or any other feedback collecting method) and make recommendations and suggestions for improvements in a separate conclusion/recommendations section.
- b. Produce a user manual (user guide) for the developed system. (This should include all the forms that developed & proper guidance with step-by-step point format.)
- c. Produce a technical documentation for the developed database system. All developed forms with source code & suitable diagrams (Use case diagram, class diagram, flow charts, DFD level 0 & 1) should be included in the technical documentation.
- d. Assess the developed database by suggesting future enhancements to ensure the effectiveness of the system.

Recommended Resources

Please note that the resources listed are examples for you to use as a starting point in your research – the list is not definitive.

Web links:

<https://support.microsoft.com/en-GB> (2022) Database design basics [online] Available at:

<https://support.microsoft.com/en-us/office/database-design-basics-eb2159cf-1e30-401a-8084-bd4f9c9ca1f5> [Accessed 1 August 2022]

<https://www.guru99.com/> (2022) Database (Data) Testing Tutorial with Sample Test Cases [online] Available at: <https://www.guru99.com/data-testing.html> [Accessed 1 August 2022]

<https://www.guru99.com/> (2022) Database Design in DBMS Tutorial: Learn Data Modeling [online] Available at: <https://www.guru99.com/database-design.html> [Accessed 1 August 2022]

<https://www.integrate.io/> (2021) Complete Guide to Database Schema Design [online] Available at: <https://www.integrate.io/blog/complete-guide-to-database-schema-designguide/> [Accessed 1 August 2022]

<https://www.lucidchart.com/pages/> (2022) Database Structure and Design Tutorial [online] Available at: <https://www.lucidchart.com/pages/database-diagram/database-design> [Accessed 1 August 2022]

<https://www.softwaretestinghelp.com/> (2022) Database Testing Complete Guide (Why, What, And How To Test Data) [online] Available at: <https://www.softwaretestinghelp.com/databasetesting-process/> [Accessed 1 August 2022]

Journal articles:

Batra, D. & Davis, J. (1992). Conceptual data modelling in database design: similarities and differences between expert and novice designers. *International Journal of Man-Machine Studies*, Volume 37, Issue 1, 1992, pp. 83-101. [https://doi.org/10.1016/00207373\(92\)90092-Y](https://doi.org/10.1016/00207373(92)90092-Y).

Gunjal, B. (2003). Database System: Concepts and Design. Proceedings of 24th IASLIC-SIG-2003.

Kaur, T. & Singh B. (2003). Testing of Databases. *IJiset - International Journal of Innovative Science, Engineering & Technology*, Vol. 1 Issue 6. ISSN 2348 – 7968

Kaur, Taranpreet & Sehra, Sumeet Kaur. (2015). Designing and Development of Database Testing Tool. International Journal of Computer Applications (0975 – 8887) Volume 120 – No.19. 14. doi: 10.5120/21334-4330.

Kraleva, Radoslava & Kralev, Velin & Sinyagina, Nina & Koprinkova-Hristova, Petia & Bocheva, Nadejda. (2018). Design and Analysis of a Relational Database for Behavioral Experiments Data Processing. International Journal of Online Engineering (iJOE). 14. 117. doi:10.3991/ijoe.v14i02.7988.

Letkowski, J. (2015). Doing database design with MySQL. Journal of Technology Research. Volume 6.

Reading:

Captain, F. (2013) Six-Step Relational Database Design™: A step by step approach to relational database design and development, 2nd edn, CreateSpace Independent Publishing Platform

Hernandez, M. (2003) Database Design for Mere Mortals: A Hands-On Guide to Relational Database Design, 2nd edn, Addison Wesley

Stephens, R. (2008) Begin Database Design W / WS (Wrox Programmer to Programmer), 1st edn, Jossey-Bass

HN Global:

HN Global HN Global (2021) Reading Lists. Available at:

<https://hnglobal.hightechglobal.com/learning-zone/reading-lists>

HN Global (2021) Student Resource Library. Available at:

<https://hnglobal.hightechglobal.com/subjects/resource-libraries>

HN Global (2021) Textbooks. Available at:

<https://hnglobal.hightechglobal.com/textbooks>

Learning Outcomes and Assessment Criteria

Pass	Merit	Distinction
LO1 Use an appropriate design tool to design a relational database system for a substantial problem		
P1 Design a relational database system using appropriate design tools & techniques, containing at least six interrelated tables, with clear statements of user and system requirements	M1 Produce a comprehensive design for a fully-functional system, which includes interface and output designs, data validations and data normalization.	D1 Evaluate the effectiveness of the design in relation to user and system requirements.
LO2 Develop a fully-functional relational database system, based on an existing system design		LO2 & LO3
P2 Develop the database system with evidence of user interface, output and data validations, and querying across multiple tables. P3 Implement a query language into the relational database system.	M2 Implement a fully functional database system, which includes system security and database maintenance. M3 Assess whether meaningful data has been extracted through the use of query tools to produce appropriate management information.	D2 Evaluate the effectiveness of the database solution in relation to user and system requirements and suggest improvements.
LO3 Test the system against user and system requirements		
P4 Test the system against user and system requirements.	M4 Assess the effectiveness of the testing, including an explanation of the choice of test data used.	
LO4 Produce technical and user documentation.		
P5 Produce technical and user documentation.	M5 Produce technical and user documentation for a fully functional system, including data flow diagrams and flowcharts, describing how the system works.	D3 Evaluate the database in terms of improvements needed to ensure the continued effectiveness of the system.

Contents

	Page
Acknowledgement	
Introduction scenario	
Activity 01	
Introduction to Database Design and Development	
Advantages of database	
Disadvantages of database	
Types of database	
Requirement	
Types of requirements	
System requirements for above scenario	
User requirements for above scenario	
ERD (ENTITY RELATIONSHIP DIAGRAM)	
History of ER diagram	
Uses of Entity Relationship Diagram	
ERD COMMON SYMBOLES	
Cardinalities	
Entity Relationship Diagram (ERD) for the given Scenario	
Attributes and entities used in this scenario	
Relational Database Design for e-space solution	
Normalization	
Data Normalization Rules	
Data redundancy and the ensuing issues	
Normal form rules:	
Benefits of normalization	
Disadvantages of normalization	
Normalization of drawn ER Diagram	
Graphic User Interface (GUI) for E-SPACE SOLUTUTION	
Wireframes	
Evaluation and effectiveness of given diagram (ERD and logical design) in terms of the identified user and system requirement.	

2. Activity

Develop a relational database system according to the entity relational diagram create SQL (Structured Query Language)

Importance if SQL (Structured Query Language)

Data types in my SQL

Data Definition language (DDL)

Data Manipulation Language (DML)

Data Control Language (DCL)

The following are some of the most regularly used SQL constraints.

Creating the relational database system according to the ERD

SUITABLE IDE FOR GIVEN SCENARIO

Examples of Integrated Development Environment (IDE)

Features of IDE (Integrated Development Environment)

SUITABLE IDE FOR GIVEN SCENARIO

Interfaces E_space solution private limited

Security mechanisms

Implementing a security mechanisms in sql server

3. Activity

Testing the database system

Objectives of test plan for E_SPACE SOLUTION

Functions of the system

The purpose of test plan

Types of testing

Levels of testing for this system

Method of testing for this system

Test plan

Testcase screenshots

TEST CASE FORM

Testing effectiveness

Feedback form

Non-technical people and some technologists provided feedback on the database solution.

Overall feedback response

Activity 4

Introduction to the system

Technical documentation

Introduction

Description:

User requirements of e-space solution

System requirements for of e-space solution

USER GUIDE

UML Design for Scenario

UML Characteristics

Use Case Diagram

USE CASE DIAGRAM FOR E-SPACE SOLUTIONS

Class Diagram

Activity Diagram

Flowchart

FUTURE ENHANCEMENT SUGGESTIONS FOR E-SPACE SOLUTION

Gantt Chart

References

Acknowledgement

I needed the assistance and advice of several reputable people to complete my task successfully. I'd like to start by thanking ESOFT for providing me with a welcoming workspace where I could finish my task. I'm very happy that the report is finished. I want to express my gratitude to Mrs. Dilishi for her helpful instructions for assignments throughout my first semester. Finally, I'd like to share my sincere appreciation to all of the family members and classmates who helped me a lot in finalizing this project within the limited time. THANK YOU SO MUCH!!!

Introduction scenario

With marking a new chapter of human history, E-Space Solutions (Pvt.) Ltd decided to colonize the planet Mars in 2040. The company decided to hire a database expert to develop a database management system of the database supports the decision to do in this project. Therefore, the Data must be accurate and consistent throughout the data lifecycle. E-Space Solutions plans to collect information from people who want to go to Mars. To colonize Mars, the company needs to think about the qualifications of volunteers and how they can contribute to the process of colonizing Mars. People who want to go to Mars will be known as "colonies". Each colonist will be identified by Mars colonization ID, first name, last name, date of birth, Qualification, age, address on Earth, gender, contact number, marital status, number of family members traveling to Mars, etc. There can only be one colonist from the same family, and the colonist can decide who will accompany him to the planet Mars. If they want, the settlers can also take their whole family with them. But they will be considered dependent on the colony. In the system, the dependents are not required to identify themselves separately, they must be identified only on the basis of their colonist. But the data of the dependents must be registered with name, date of birth, age, gender, relationship of the colonist etc. E-Space Solutions plans to develop a spacecraft called E-Jets. These E-Jets take the colonists and their families to Mars and the pilots return with empty E-jets.

Activity 1

Introduction to Database Design and Development

Database design and development is the process of creating high-performance databases that can store, retrieve, and manage data. This process includes determining data requirements, designing the database structure, installing and updating the database, and applications that can access and modify the data.

Identifying the data requirements for a project or organization is the first step in database design. Considering the required data types, their relationships, and data constraints such as data types and data validation rules are all essential elements of this. Then, the database data structures such as tables, fields, and relationships between tables are developed. The structure must be designed to allow for rapid data retrieval and writing while ensuring data integrity and security.

After creating the database structure, the next step is to develop the database design and populate it with data. To do this, you must create tables, establish their links, and enter data into the tables. The final step in the process is the development of programs and interface software that allow users to access the data and perform tasks with it. This can include developing user interfaces that allow users to manipulate the data, as well as developing applications that can collect and modify data. Throughout database design and development, data management and security standards within the database must be adhered to. Data security, confidentiality, and accessibility are ensured through regular backups, data encryption, and access control.

Efficient database design and development improves a company's profitability, decision-making, and efficiency. Proper organization of the database increases the efficiency of data retrieval and storage while maintaining data accuracy and consistency.



Figure 01 Introduction to Database Design and Development

(coach, 2021)

Advantages of database

- Faster searching of data.
- Data is more clear (No hand written data)
- less room space to keep data
- efficient backup method (cloud storage, automatic backup)
- can manage user effectively with user access rights (update ,delete ,read , Permissions).
- Can access data by many users at a time using the network.

Disadvantages of database

- Complex (Need technical knowledge) to set up (create)
- user must be trained
- higher initial cost(buying server ,software ,training)
- more security threats are expected (hacking virus)
- need database administrator (salaries)

Types of database

Electronic database concept was developed step by step into the today level through different database models they are

1. flat file database
2. hierarchical database
3. network database
4. relational database
5. object relational database

1. flat file database

Is the primitive basic database model introduce by lomoago.it maintains only one data table to the entire database .it create many issues such as

- **data redundancy:** unnecessary data replication so that same data are repeated maintained within the table it leads to waste of storage space and wasting data entry time while updating of data has to be done manually by searching all the areas where the specific data are mentioned.

One value is missed used to update the database maintain wrong values in completed. Maintain a single table create many issues therefore the solution is to maintain multiple table to manage within database

EMPNO	NAME	UNIT	JOB CODE	LEVEL	TITLE	SEX	BIRTH DATE	PRIMARY SKILL	SECONDARY SKILLS	SALARY
45584	PETERSON, N.M.	2000	0110	HEAD	DIVISION MANAGER	M	280607	0110	6130 6625	6040
32579	LYNN, K.R.	2000	5210	EMPL	SECRETARY	F	530121	5210	5520	12000
57060	CARR, P.I.	2100	1110	HEAD	MANAGER DEVELOP DEPT	M	350720	1110	1130 1135	0130 1355
15324	CALLAGAN, R.F.	2100	5210	EMPL	SECRETARY	F	550606	5210	5520 5220	10800
10261	GUTTMAN, G.J.	2110	1110	HEAD	MANAGER SYSTEMS GROUP	PM	301110	1110	1130 1135	0150
72556	HARRIS, D.L.	2110	5210	EMPL	SECRETARY	F	550517	5210	5520	8400
24188	WALTERS, R.J.	2111	1110	HEAD	CHIEF PROPOSAL SECTION	M	260202	1110	1120	28000
21675	SCARBOROUGH, J.B.	2111	1120	EMPL	MECH ENGR	M	240914	1120		21000
18130	HENDERSON, R.G.	2111	1130	EMPL	ELEC ENGR	M	340121	1130		23000
91152	GARBER, R.E.	2111	1130	EMPL	ELEC ENGR	M	440707	1330	1130	16400
30793	COMPTON, D.R.	2111	1350	EMPL	COST ESTIMATOR	M	290328	1350	1351 1355	1130
81599	FRIEDMAN, J.M.	2112	1110	HEAD	CHIEF DESIGN SECTION	M	360317	1110	1130	26000
21777	FRANCIS, G.C.	2112	1110	EMPL	SYSTEMS ENGR	M	321111	1110	1130	24000
24749	FAULKNER, W.M.	2112	1120	EMPL	MECH ENGR	M	400621	1120	1130 1330	24000
13681	FITINGER, G.J.	2112	1130	EMPL	ELEC ENGR	M	431216	1130	1355	22000
82802	APGAR, A.J.	2112	1130	EMPL	ELEC ENGR	M	500715	1130	1330	21000
63633	BLANK, L.F.	2112	1330	EMPL	DRAFTSMAN	F	491010	1330		16000
22959	BRIGGS, G.R.	2115	1110	HEAD	CHIEF PROD SPEC SECTION	NM	400508	1110	1120	24000
29414	ARTHUR, P.J.	2115	1120	EMPL	MECH ENGR	M	300109	1120		22000
37113	ARNETTE, L.J.	2115	1130	EMPL	ELEC ENGR	M	450729	1130		22000

Figure 02- flat file database

(Everest, 1973)

2. hierarchical database

Hierarchical database model is maintain different tables to represent the database while they are connected related to one another by forming hierarchical tree structure. When maintain the hierarchical connectivity on the top to bottom (parent child relationship) can be maintained but a practical database they are requirement to violate the top to bottom arrangement

Hierarchical database are supporting one to one, one to many relationship but not many to many relationship.

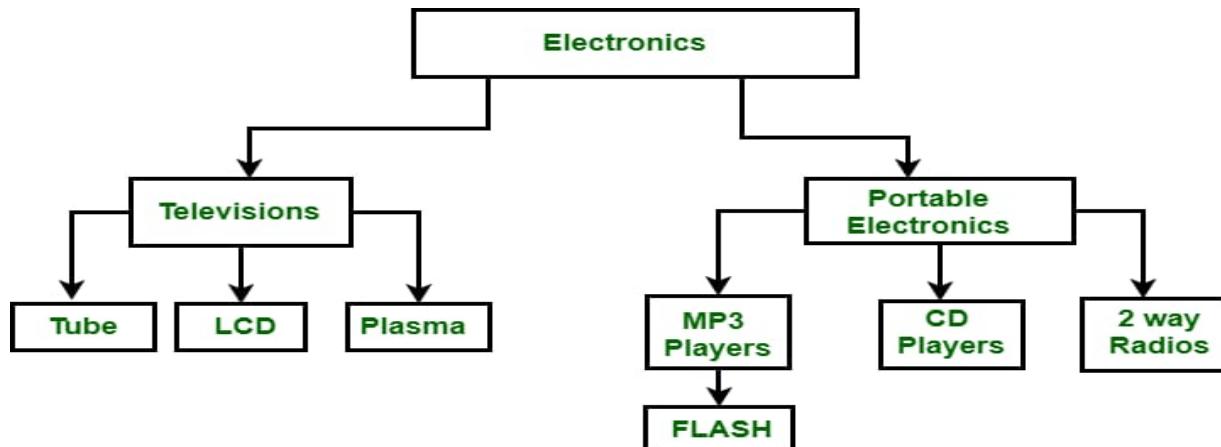


Figure 03 hierarchical database

(itskawal2000, 2022)

3. Network database model

the problem of maintain the hierarchical order tree is solved so the data table can be related to one another in any way where they required but the connectivity are established based on the each record for separately manually

Creating a manual connectivity among the record will make a mistake since it is done by data entry operators (connecting to wrong records, misuse to connect with required records)

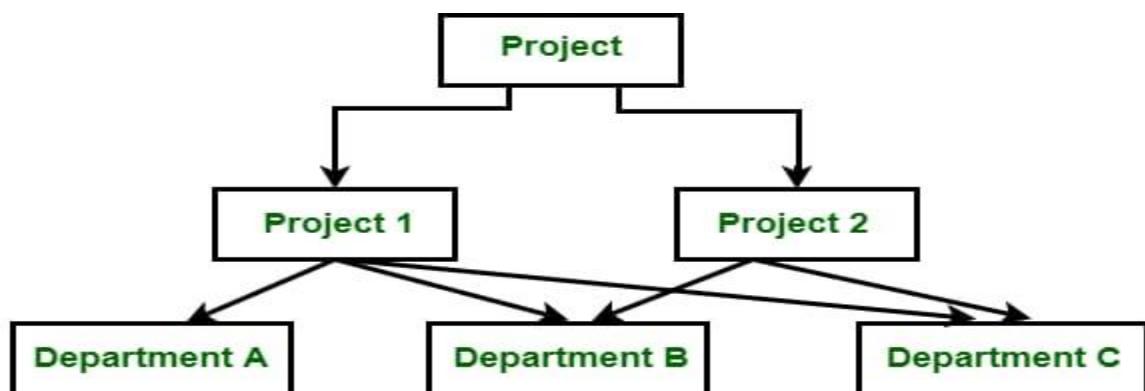


Figure 04 network database model

(itskawal2000, 2022)

3. Relational database model

As successful database model which is still used in the modern society the data are maintained in different table while they are successfully connected to make a meaningful data they are keywords related database will primary key, record, fields

It is not possible to maintain duplicate value under the primary key column relational database model maintains many different numbers of table which are independent to one another but they are related connected to another using foreign key

People	Name	Gender	Coffee Drinker
	Anna	F	T
	Sam	M	F
	Bob	M	F

Friend	Name1	Name2
	Anna	Sam
	Sam	Anna
	Anna	Bob
	Bob	Anna

Figure 05 relational database model

(schulte, n.d.)

6. object oriented database

The existing relational database concepts are added with object oriented programming concept in order to constraint efficient successful database application is it is a future database technologies but it is still in use in some application(extended version of the relationship) Electronic database are maintaining a client server model so that all the data tables records are available in the central database server while using client accessing reading updating data through the network

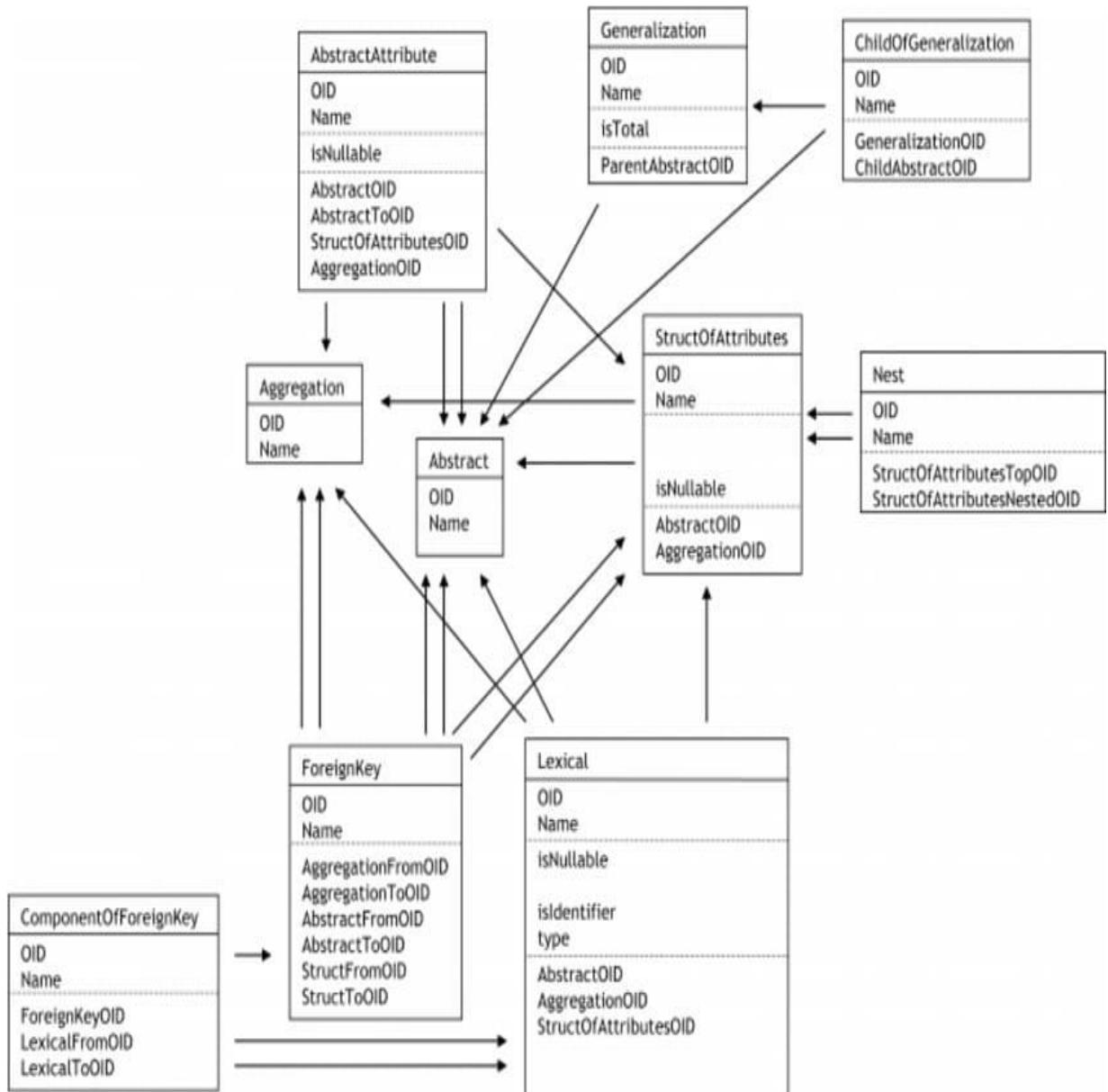


Figure 06 object oriented database

(atzeni, 2009)

Requirement

The requirements describe the users' expectations for the software product are communicated through the requirements. The client's requirement may be clear or unclear, known or unknown, and planned or unplanned.

- **Functional Requirements**

Functional requirements are a set of guidelines that define the necessary features, abilities, and interactions that a database system must possess to satisfy the requirements of its users.

- **Non-Functional Requirements**

Non-functional requirements for a database encompass the characteristics and qualities that define the overall behavior, performance, and usability of the database system. Unlike functional requirements that outline specific features and functionalities, non-functional requirements concentrate on aspects like,

- Performance The performance of a database is all about how efficiently and quickly the system works, especially when it comes to running queries, handling

Types of requirements

In the database the requirement are divided in to 2 groups. They are

- System requirement
- User requirements

➤ System requirement

The system requirements for hardware or software to function properly are called system requirements. If these requirements are not met, installation or performance issues may occur. The former may block the installation of a device or application, while the latter may cause the product to malfunction, freeze, or crash.

In the system requirement their can classified furthermore there are

- ❖ Software specification
- ❖ Hardware specification

First we will discuss about software specification

❖ **Software specification**

Software specification outlines the functions and performance standards for the software.

Ex;

Operating System,
Microsoft Data Access Components,
SQL Server 2014 Management Studio
Oracle
Visual studio code 2019, pycham

❖ **Hardware specification**

Requirements of hardware devices. Most hardware mainly has compatibility or operating system requirements.

Ex;

Processor: Intel Core (TM) i7-8550U CPU Processor 8 Cores 16 Threads
(16M Cache, up to 5.10 GHz)

RAM: Corsair Vengeance LPX 32GB DDR4 SDRAM (2 x 16GB)
3200MHz Desktop Memory

Storage: Samsung 970 EVO Plus 1TB NVMe SSD

Network: 1Gbps Ethernet LAN Card and Wi-Fi

Backup and Recovery: Synology DS420j 4-Bay NAS Enclosure with Seagate Iron Wolf 8TB NAS Internal Hard Drive.

System type: 64-bit operating system, x64 based processor

Other hardware: printers/scanners, UPS

➤ **User requirements**

User requirements are the functionalities that end users of a particular system can use. These are focus on the needs and expectations of users. It can include functional requirements like inserting, updating, and deleting as well as non-functional requirements like security, performance, and flexibility.

Building adaptive reuse, in the context of a pandemic is typically seen as the reuse of an existing building in a manner that retains as much as possible of the original building while updating its performance to meet modern codes, standards, and new user requirements.

System requirements for above scenario

1. the database should able to collect and maintain data on colonists, dependents, e-jets ,trips and houses and should be save by the system
2. maintain high security on the data base and information
3. training on generating and interpreting reports
4. maintain the data accuracy and consistency and reduces the data redundancy throughout the whole database

User requirements for above scenario

1. the database should be able to store information efficiently and maintain comprehensive data on colonists, dependents, e-jets, trips and houses
2. The database should allow for the creation of multiple passengers for a single trip as well as each house is assigned to a colonist.
3. The database should have security measures in place to protect sensitive information.
4. The database should have the ability to back up data regularly to prevent loss of information.
5. The database should be user-friendly and easy to navigate.
6. The database should have the ability to generate reports on the information such as Jet Detail report, Trip Detail report with passengers, Colonists details with dependents & etc., as needed.
7. The database should maintain accurate information about trips and passengers details.
8. The database should have an efficient data retrieval and storage method.

9. The database should be able to store information about each colonist , including a name and any relevant details as well as The pilots can check the data of the system according to the space craft trip information with the passenger details
10. The database should have high security in order to protect the colonists details
11. When the database ongoing the Colony Superintendent will generate Colony report with House details & colonists, Job reports of the colonists & etc.
12. The system should be able to create, add, edit and delete information using the system.

ERD (ENTITY RELATIONSHIP DIAGRAM)

An entity relationship diagram is also called as ER diagram and entity relationship models.ER diagram is a type of flowchart that visualizes how the relationship between “entities” such as people, objects or concepts relate to each other within a system.

History of ER diagram

In 1976, Peter Chen presented a paper titled "The Entity-Relationship Model: Toward a Unified View of Data," outlining his idea for entity relationship modelling. Chen had developed ERDs during the 1970s. Computer scientist Peter Chen worked to enhance database design. His entity relationship model provided a means of visualizing a database that brought together many models that were already in use to eliminate confusion. The network model, relational model, and entity set model were the three types of database data models used before ERDs. While each had advantages and disadvantages of its own, none offered a comprehensive picture of the database. Chen might offer a single paradigm for database modelling with an ERD.

The scientists and engineers who came before Peter Chen, particularly Charles Bachman, who worked on database visualization in the 1960s and whose data structure diagrams became known as Bachman diagrams, had a significant impact on Chen's work.

In many respects, Chen's entity connection model served as the paradigm for later information systems methods such as Unified Modelling Language, or UML.

James Martin, a different computer scientist, developed Chen's ER model further in the 1980s and introduced what is now called the IE notation. Instead of using Chen's notation to indicate cardinality (one to many relationships), IE notation employs Crow's foot. (S, 2024)

Uses of Entity Relationship Diagram

- Database design
- Database troubleshooting
- Business information systems
- Business process re-engineering (BPR)
- Education
- Research

ERD COMMON SYMBOLS

Entities

Entity is defined as any real world object .Entity is anything in the enterprise that needs to be represented in our database; it might be a physical object or an event that occurs in the actual world. Every entity is made up of several properties that represent that entity. It can be labelled as **Rectangle**

Entities can be in many different categories

Human: Student, teachers, employee, manager

Non-living: Item, vehicle, books

Intangible: Subject, bank account

Place: city, town

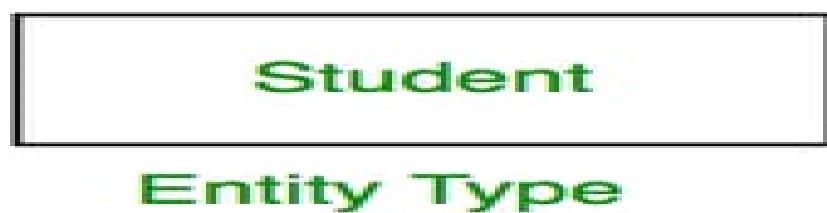


Figure 07 Entities

(codersaty, 2021)

The entity is: student

Attributes

An entity's attributes are its properties or characteristics. Each attribute has a name and is linked to an entity. Are data items to describe the entities in details and which should be connected to the specific entity where it describes? And represented by label oval

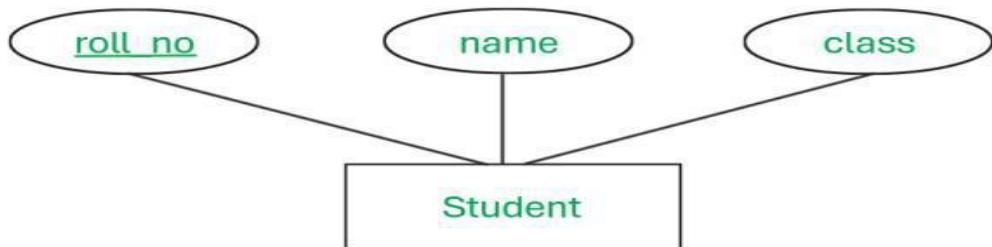


Figure 08 Attributes

(mangalgiaishwarya2, 2024)

The attributes are: roll no, name, class

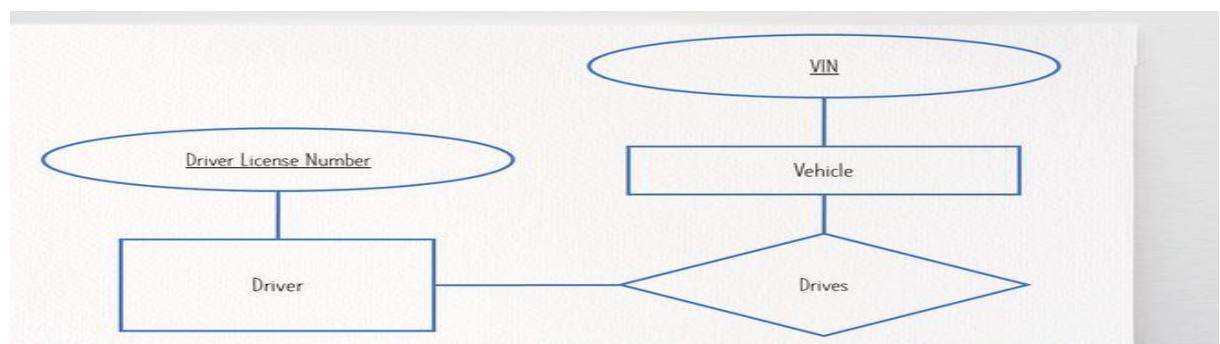
There are many types of attributes, but we mainly focus on 4 attributes such as

- primary key attributes
- composite attributes
- multi valued attributes
- derived attribute

First discuss about

❖ primary key attributes

when ER diagram is converted into table format there must be a primary key so that the specific primary attributes ER diagram must be **underlined** if it is a composite key (more than one column for the primary key)the necessary attribute underlined



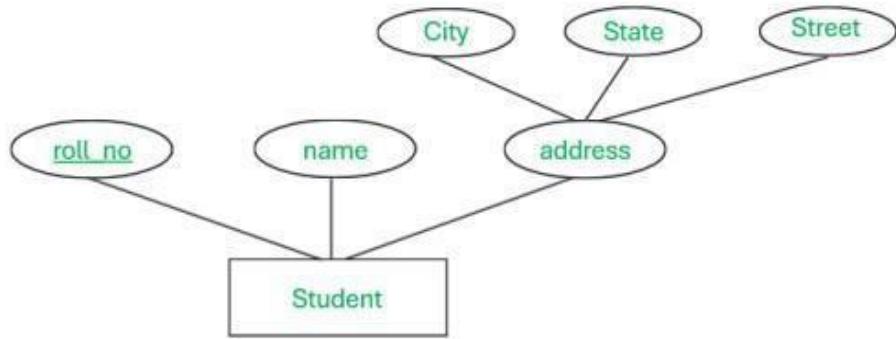
❖ Figure 09 - primary key attribute

(glen, 2021)

The primary key is: driver license number, VIN

❖ composite attributes

This is a special attribute which is made with compose or can be split into components of (some more attributes)



❖ *Figure 10 - composite attributes*

(mangalgiaishwarya2, 2024)

The composite attributes are: city, state, street

❖ multi valued attributes

An another attributes where more than one value can be maintained under a column for single record it maintain multiple value for single instance

Multi valued are represented by using a double line oval attribute

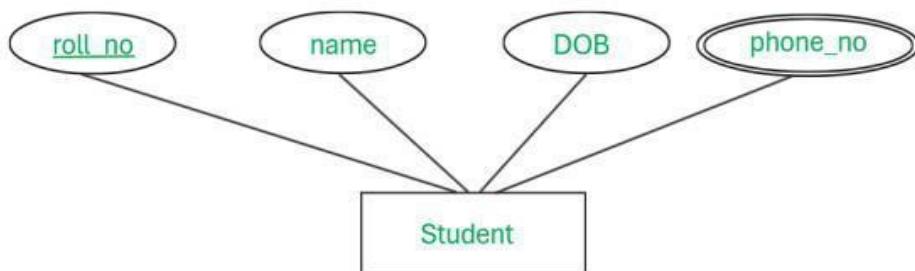


Figure 11 - multi valued attributes

(mangalgiaishwarya2, 2024)

The multi valued attributes are: phone no

❖ derived attribute

Is an attribute associated with an entity where the corresponding data for the specific attribute can be automatically calculated with existing data in the table attributed to presented by using dotted line over

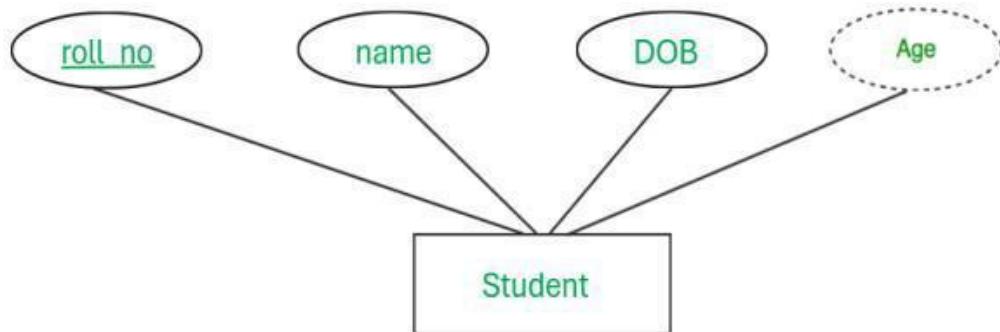


Figure 12 - derived attribute

(mangalgiaishwarya2, 2024)

The derived attributes are: age

Relationships

Are representing the association among the entities which is represented using **diamond**. The entity is used for the above relationship must be connected in the ER diagram. Relationship can be easily identified by focusing onto the verb in the paragraph

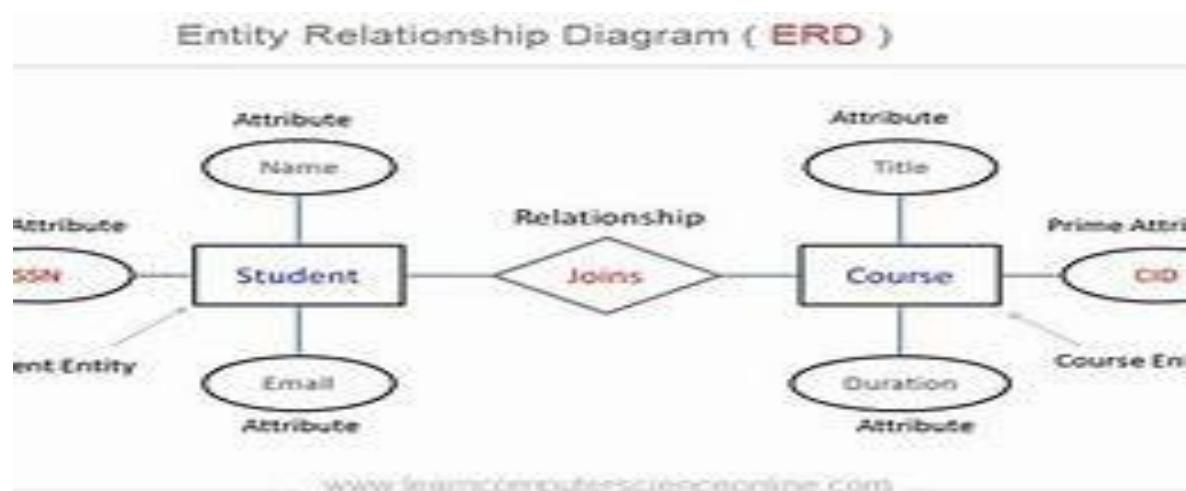


Figure 13 - Relationships

(KUMARI, 2024)

- When two entities are connected through a relationship it is called as **binary relationship**.
- When they are more than two entities connected through the relationship it is called as **non-binary relationship**.

Cardinalities

Cardinality refers to the number of relationships that exist between any two entity sets in a relationship model. There are three types of cardinality:

1. one-to-one
2. one-to-many
3. many-to-many

One-to-one

a single instance of the entity A is exactly matching with only one instance of the entity B.while a single instance of entity B is exactly matching with only one instance of their entity A

One-to-many

A single instance of entity A is associated with more than one instance of the entity B while a single instance of the entity B is associated only one instance of the entity A

Many-to-many

a single instance of the entity A is associated with many instance of the many instance of the B while a single instance of the B is associated with many instances of entity A.

Entity Relationship Diagram (ERD) for the given Scenario

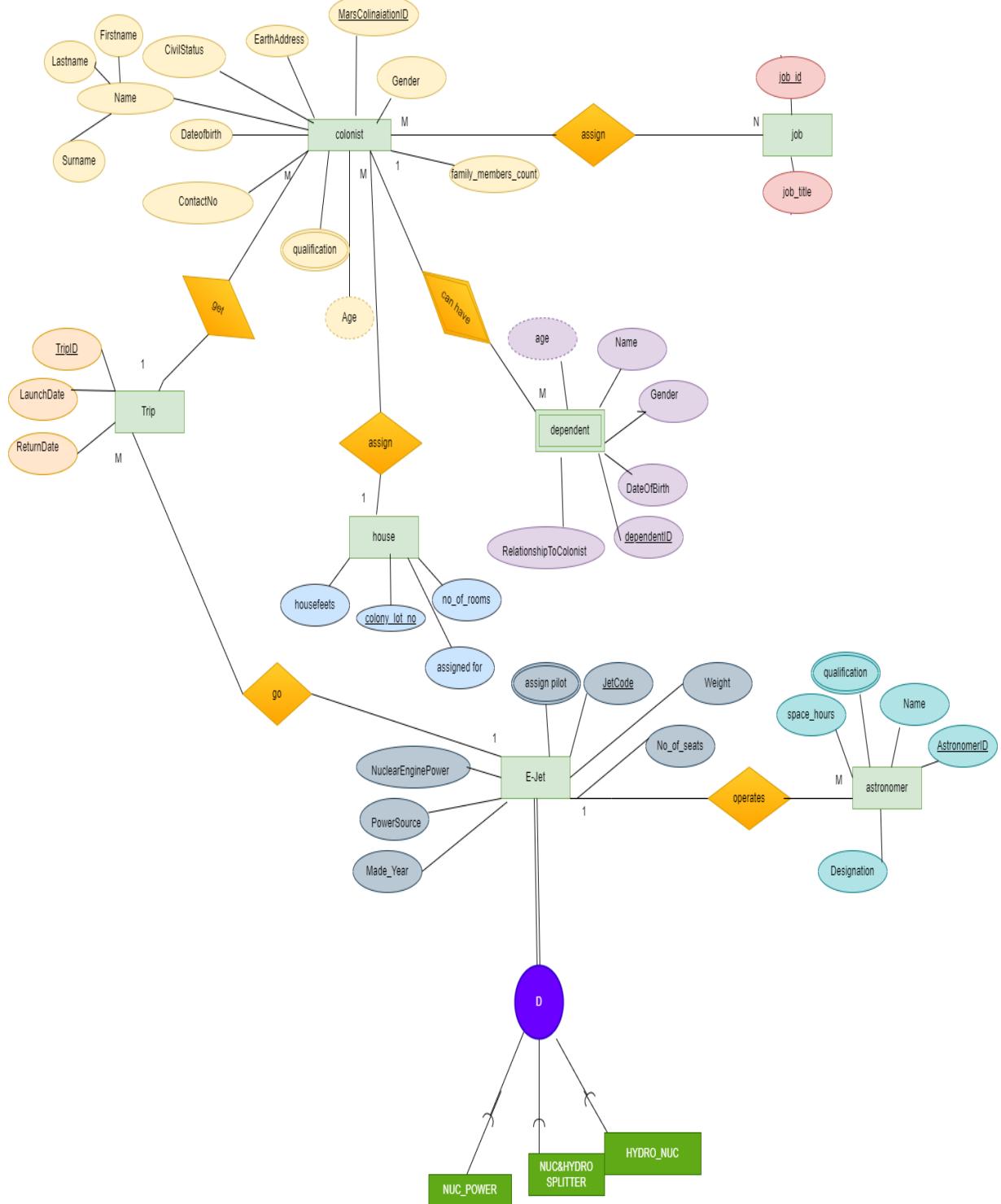


Figure 14 Entity Relationship Diagram (ERD) for the given Scenario

Attributes and entities used in this scenario

Entity	Attribute
❖ Colonist	MarsColonizationID (PK) FirstName MiddleName Surname DateOfBirth Qualification Age EarthAddress Gender Contact_No Civil Status Number Family Members
❖ Dependent	DependentID (PK) ColonistID (FK) FirstName DateOfBirth Age Gender Relationship to colonist
❖ E-Jet	Jet Code (PK) Type (Nuclear, Hybrid, Hydro-Nuc) Passenger Seats NuclearEnginePower Made Year Weight Assigned pilot Power Source
❖ Trip	Trip ID (PK) Jet Code (FK) Launch Date Return Date

❖ House	ColonyLotNumber (PK) NumberOfRooms Square Feet Assigned for Colonist ID (FK)
❖ Job	Job ID (PK) Job type
❖ Colonist Job	Colonist ID (FK) Job ID (FK)
❖ Astronomer	Astronomer ID (PK) Name Qualification Designation Space hours Rank

Cardinalities

- Colonist and Dependents - one to many
- E-Jet and trip - one to many
- Colonist and Jobs - many to many
- Colonists and house – one to one

Assume : one to Many relationship if it shared

- Trip and colonist – many to many
- E-jet and astronomers –one to many

Relational Database Design for e-space solution

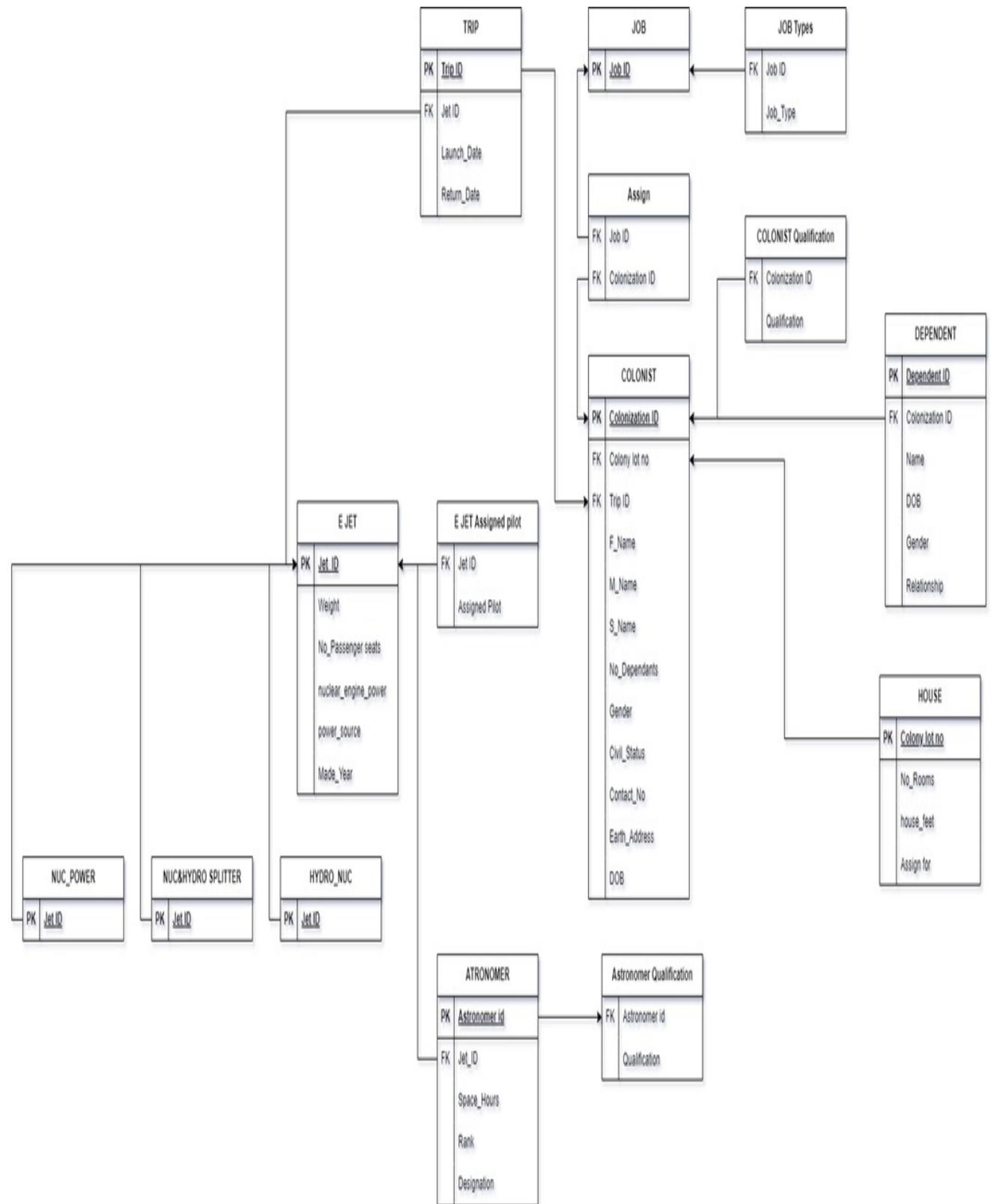


Figure 15 Relational Database Design for e-space solution

Normalization

Database normalization is the process of organizing data into tables so that the results of using the database are always clear and consistent with expectations. Such normalization is the essence of relational database theory. It can have the effect of duplicating data within the database and often results in the creation of additional tables.

The concept of database normalization is often attributed to E.F. Cod, an IBM researcher who published a paper describing the relational database model in 1970. What Cod described as a? Normal form for database relations? is an essential part of relational engineering. This data normalization gained traction in the 1970s and 1980s, when hard drives were expensive and high-performance data storage was essential. Since then, other techniques, especially denormalization, have also gained popularity. (Vaughan, 2024)

Data Normalization Rules

Although data normalization rules tend to increase data duplication, they do not create data redundancy, which is unnecessary duplication. Database normalization is generally a refinement process after the initial exercise of identifying the data objects that should be in a relational database, defining their relationships, and defining the required tables and columns for each table. (Vaughan, 2024)

Data redundancy and the ensuing issues

- Expand the database's size
- Add anomaly (problems)
- Anomaly of deletion (Problems)
- Anomaly in updating (Problems)

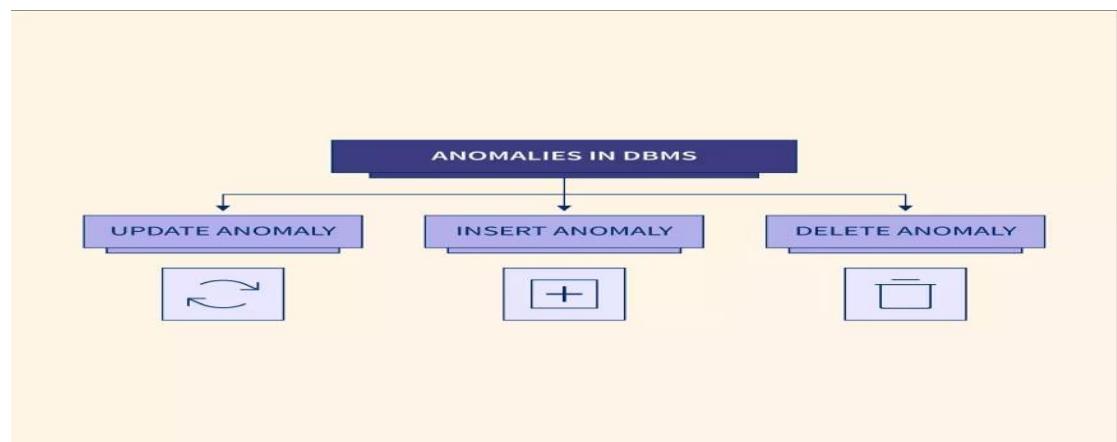


Figure 16

(venkatachalam, 2024)

There are 4 types of normalization forms

1. First normal form (1NF)
2. Second normal form (2NF)
3. Third normal form (3NF)
4. Boyce cod normal form (BCNF)

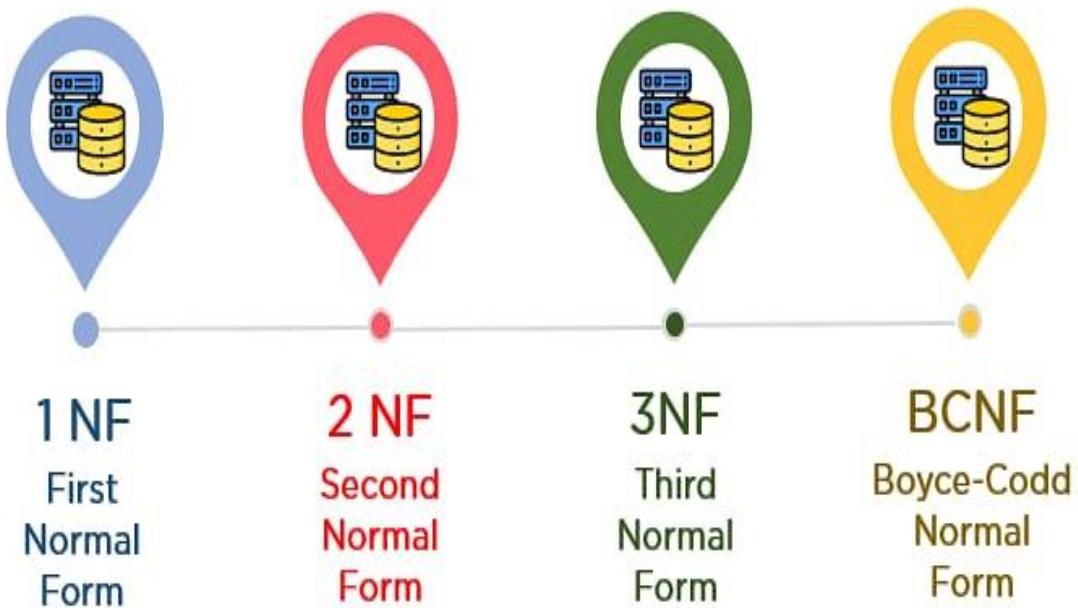


Figure 17 There are 4 types of normalization forms

(S, 2024)

Normal form rules:

- Only unique-valued attributes.
- The attribute domain does not change.
- Each attribute/column has a unique name.
- It doesn't matter in what order the data is stored

Benefits of normalization

- ❖ Normalization reduces data redundancy.
- ❖ Better overall database structure.
- ❖ Database data consistency.
- ❖ Database design is far more adaptable.
- ❖ Reinforces the principle of relationship integrity.

Disadvantages of normalization

- ❖ Once you understand what the user requires, you cannot begin to develop the database.
- ❖ A poorly designed database can result from careless deconstruction, which might cause major issues.
- ❖ Normalizing interactions at a greater degree takes a lot of time and effort. When the relations are normalized to higher normal forms, the performance deteriorates.

First Normal Form (1NF)

Relation has any repeating groups (Multi values and composite attributes)it should be remove and establish new ones.

Steps from UNF (Unnormalized Form) to 1NF:

- Remove the repeating group from the relationship and form a new one.
- Include a copy of the primary key from the original relationship.
- Determine the foreign key and rename, repeat the steps no more repeating groups.

Second Normal Form

A relation must be in first normal form to be in second normal form, and it must not have any partial dependency. If a relationship has No Partial Dependence, it is classified as 2NF.

Steps from 1NF to 2NF

- Remove the partial dependency and create a new relation.
- Add a copy of determinant and that will automatically become a foreign key of the new relation.
- Name the new relation and rename the original relation.

Third Normal Form (3NF)

If the relation has any transitive dependency (indirect connection between two unimportant characteristics), It should remove and create a new relation.

Steps from 2NF to 3NF

- Remove the transitive dependency and create a new relation.
- Add a copy of determinant and that will automatically become the primary key of the new relation.
- Name the new relation and rename the original relation

Boyce-code Normal Form (BCNF)

The relation is in BCNF if and only there is a super key within the relation and It is also known as 3.5 Normal form.

Normalization of drawn ER Diagram

Ensure that the database is in at least third normal form (3NF):

1NF-each table has a primary key, and all attributes are atomic.

2NF-all non-key attributes are fully functional dependent on the primary key.

3NF-no transitive dependency (non key attributes do not depend on other non-key attributes)

Colonist table

Mars Colonization ID	Name	Date Of birth	qualification	age	Earth address	gender	contact number	civil status	Family Members Count

- Already in 3NF; no repeating groups, no partial dependencies, and no transitive dependencies.

Dependent table

Dependent ID	Colonist ID	Name	DateOfBirth	Age	Gender	RelationshipToColonist

- Already in 3NF; no repeating groups, no partial dependencies, and no transitive dependencies.

E-Jet table

Jet Code	NumberOfSeats	NuclearEnginePower	Made Year	Weight	Power Source

- Already in 3NF; no repeating groups, no partial dependencies, and no transitive dependencies.

Astronomer table

Astronomer ID	Name	Qualifications	Designation

- Already in 3NF; no repeating groups, no partial dependencies, and no transitive dependencies.

Trip table

TripID	Jet Code	Launch Date	Return Date

- Already in 3NF; no repeating groups, no partial dependencies, and no transitive dependencies.

House table

ColonyLotNumber	NumberOfRooms	Square Feet

- Already in 3NF; no repeating groups, no partial dependencies, and no transitive dependencies.

Job table

jobID	Jobtitle

- Already in 3NF; no repeating groups, no partial dependencies, and no transitive dependencies.

ColonistJob table

ColonistID	jobID

- Already in 3NF; no repeating groups, no partial dependencies, and no transitive dependencies.

HouseAssignment table

ColonistID	ColonyLotNumber

- Already in 3NF; no repeating groups, no partial dependencies, and no transitive dependencies.

ColonistTrip table

TripID	ColonistID

- Already in 3NF; no repeating groups, no partial dependencies, and no transitive dependencies.

Graphic User Interface (GUI) for E-SPACE SOLUTION

There are eleven interfaces available from the e-space solution. The login page is the initial interface we view when we log in to the system. With this, one of us may log in with the username and password to visit the system's main page or home page. The main menu offers eight options to select from: Colonist, dependent, e-jet, trip, house, job, colonist job, astronomer and Settings. You can explore within that selection by selecting one. To go back to the login page, you may alternatively click the back button. After choosing an option, information about it may be viewed. The database records of those records can then be updated, deleted, clear or inserted.

Wireframes

The most basic visual representation of your concept is a wireframe. It is a two-dimensional schematic representation of an interface that is mostly intended to convey the layout's structure, including the parts that it will contain and their relative placement. In the ideation stage of the design process, when a team is exploring different design approaches, wireframes are typically employed early in the process.

1. Login page for e-space solution.

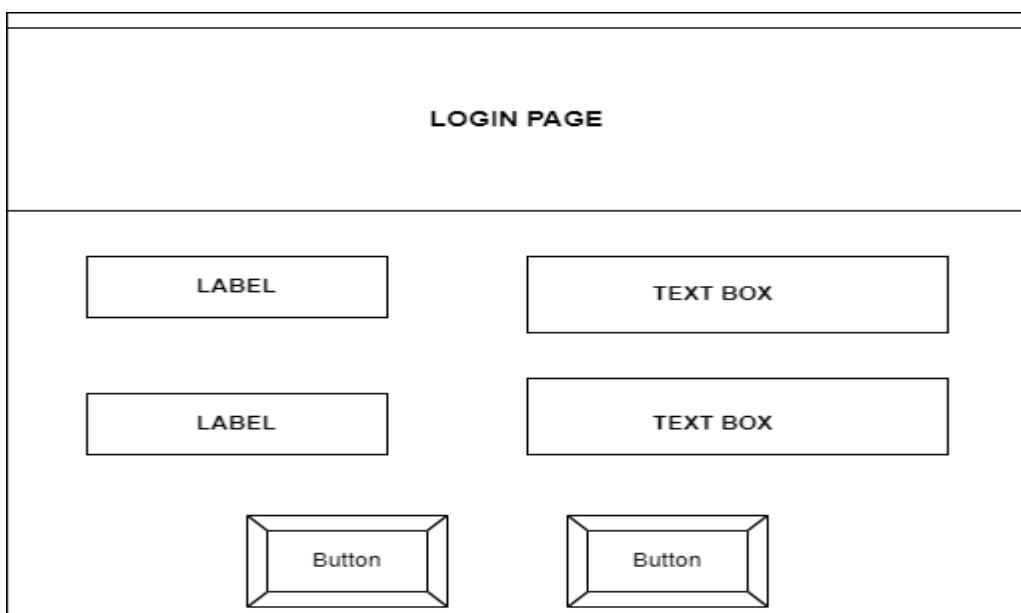


Figure 18 - Login page for e-space solution.

2. Home page for e-space solution.

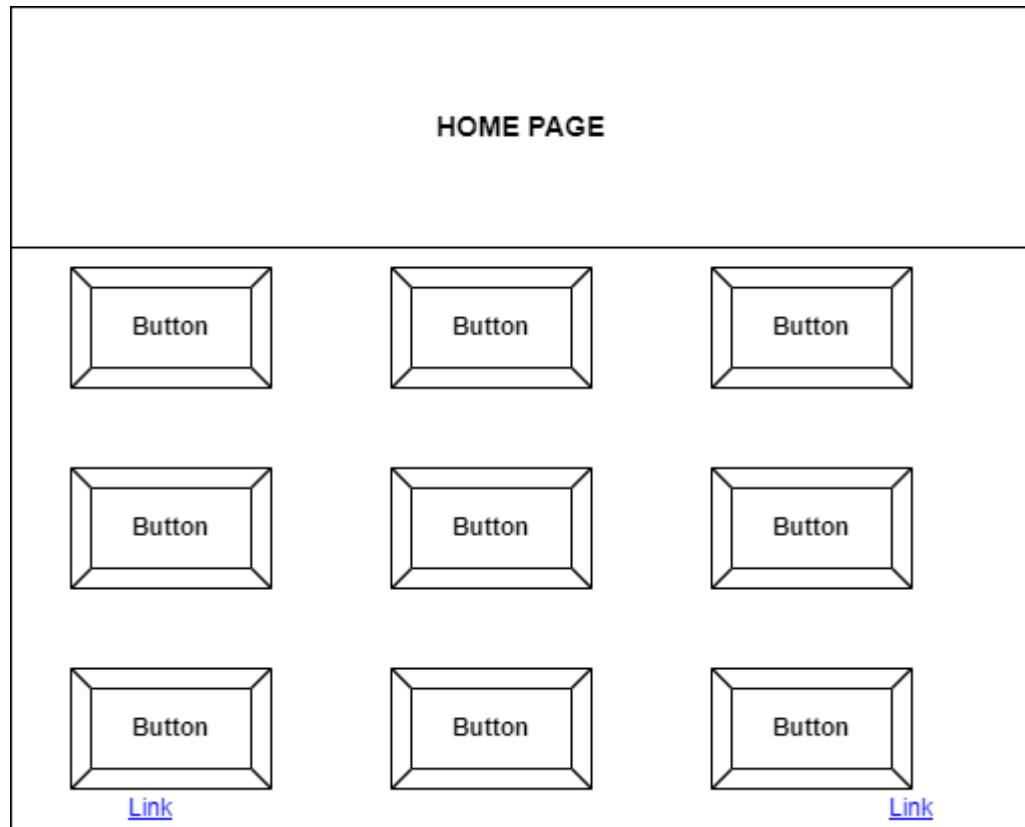


Figure 19 - Home page for e-space solution

3. Colonist interface for e-space solution

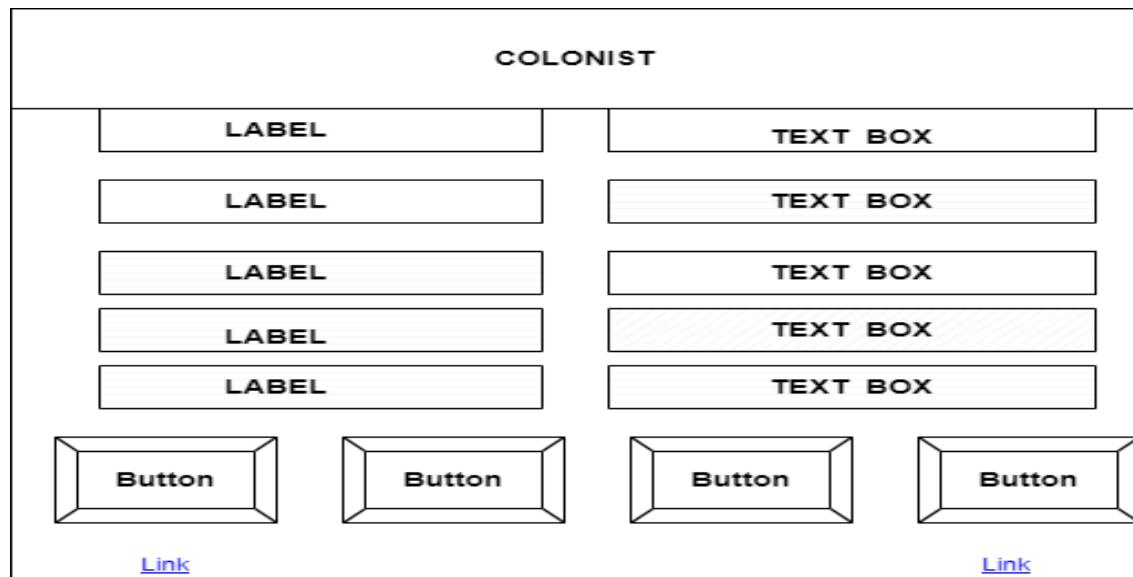


Figure 20 - Colonist interface for e-space solution

4. Dependent interface for e-space solution

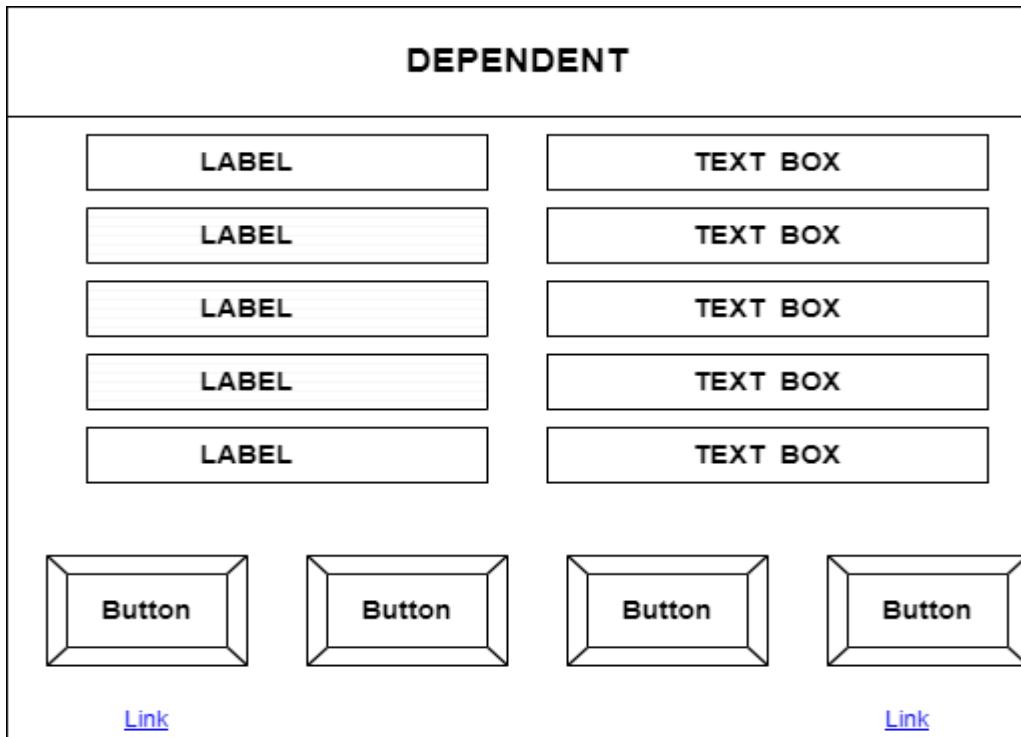


Figure 21- Dependent interface for e-space solution

5. E-jet interface for e-space solution

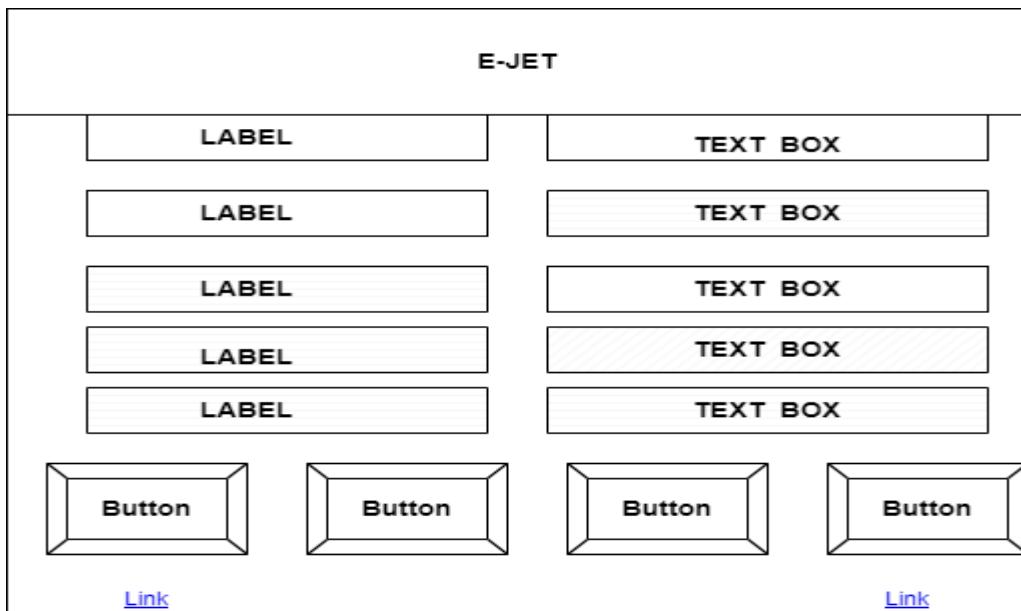


Figure 22 - E-jet interface for e-space solution

6. Astronomer interface for e-space solution

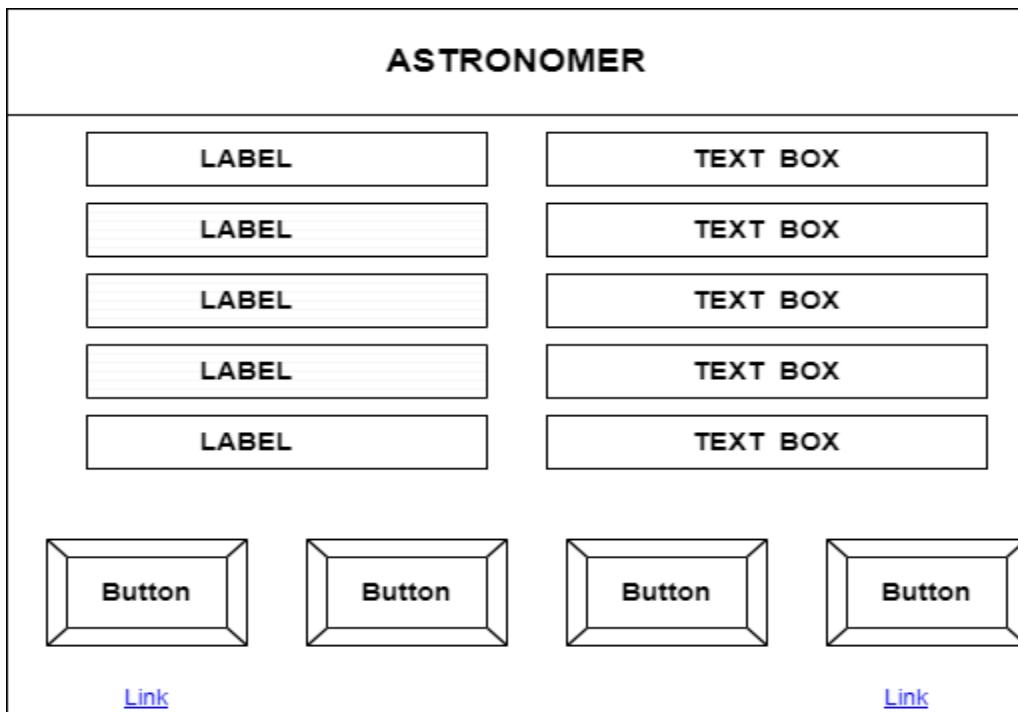


Figure 23 -Astronomer interface for e-space solution

7. House interface for e-space solution

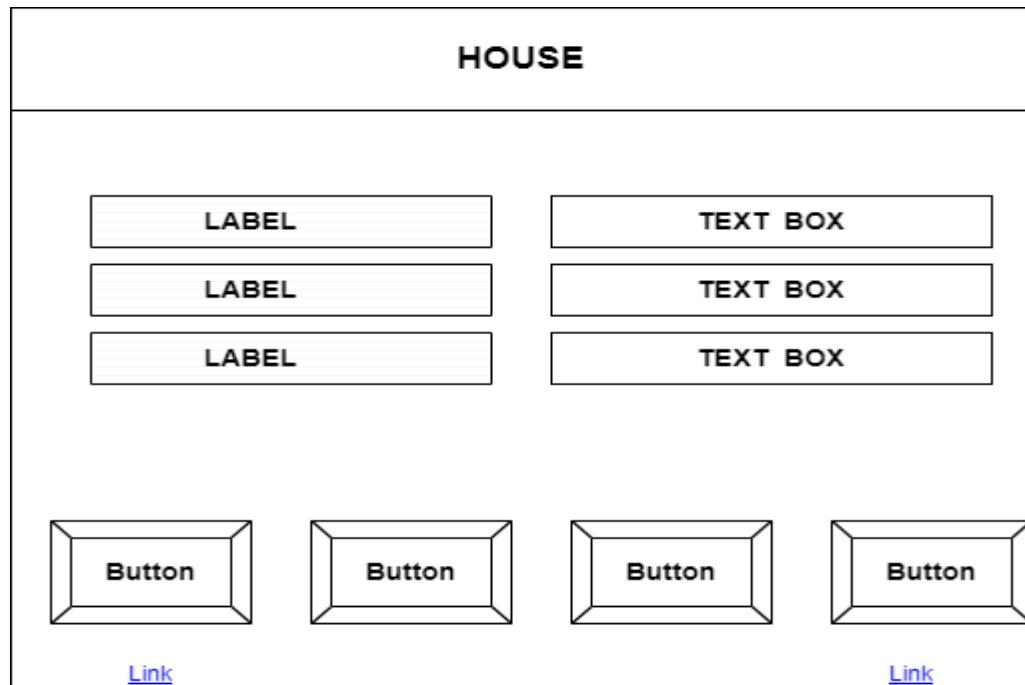


Figure 24 - House interface for e-space solution

8. Job interface for e-space solution

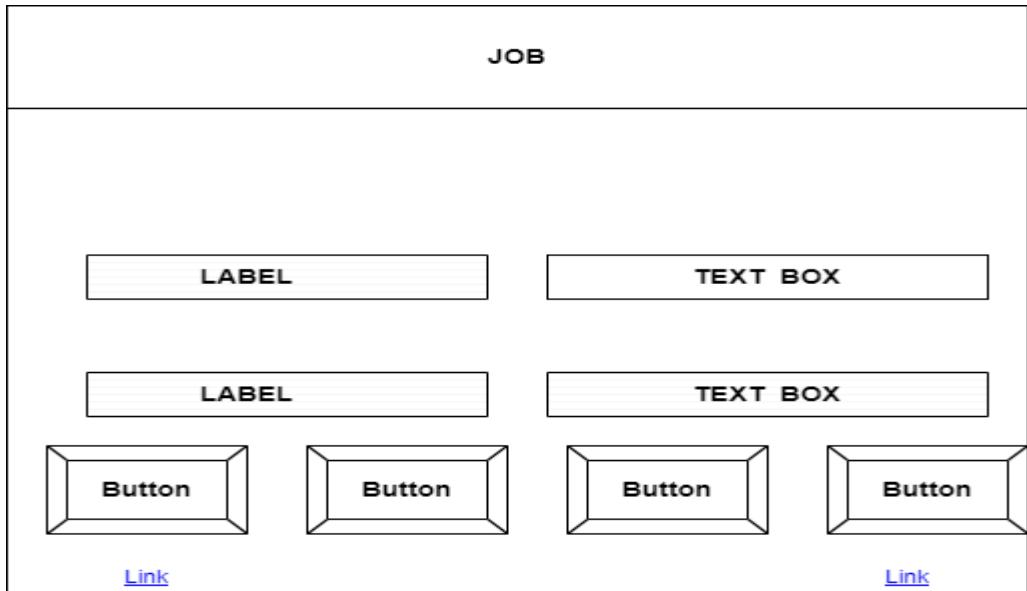


Figure 25- Job interface for e-space solution

9. Trip interface for e-space solution

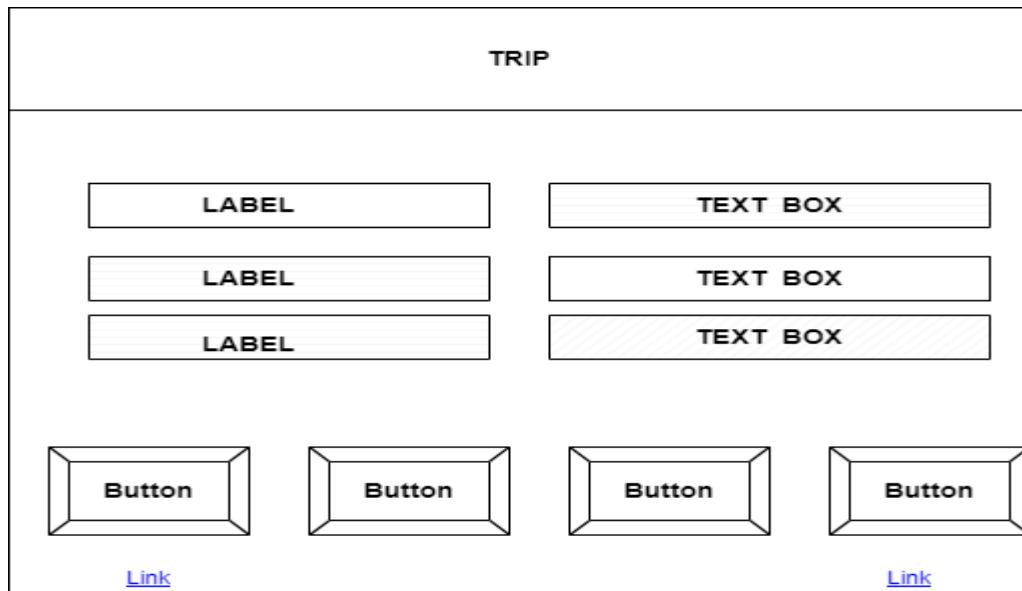


Figure 26- Trip interface for e-space solution

Evaluation and effectiveness of given diagram (ERD and logical design) in terms of the identified user and system requirement.

- The given design is effective in meeting the identified user and system requirements for E-Space solution because it incorporates both of the ERD and the logical design here are some of the reason.
- ❖ After creating the entity relationship diagram, I generated this database. It is quite impossible to create a database without using ER diagrams. If there are any flaws in the ER diagram, the database is wrong. After developing the ERD, the following step is to generate and normalize tables. The normalized relations are then used to build the relational schema. I used Draw.io to design my ERD and relational structure.

Advantages of drawing an ER Diagram,

- Easy to understand
- Popular
- Simple
- Good collection of theories and good practices
- More specific to relational database modelling
- Good DBMS support
- Good support from providers and tools
- ❖ The ERD effectively represents the relationships between the system's various entities in the system, such as colonist, dependent, e-jet, trip, house, and job. This makes it easier to understand how different data related to each other and how they can be accessed.
- ❖ The user requirements are effectively met by a set of simple user interfaces designed to input and output data from the database. The interfaces are simple to use and understand, allowing for efficient of data.
- ❖ The ERD is effectively translated into a set of tables that can be used to store and manipulate data by the logical database design. Primary and foreign keys are used in the design to ensure data integrity and consistency between tables.
- ❖ The design meets the needs of the users by allowing the storage and management of colonist, dependent, e-jet, trip, house, job and colonist-job data, as well as the ability to assign data to that.

- ❖ The system requirements are also met since the design is scalable and adaptable to changes or additions to the data being stored. Furthermore, the design is normalized, which aids in the reduction of data redundancy and the prevention of anomalies such as update, insertion, and deletion anomalies.

- ❖ Covers all identified entities and relationships: The design incorporates all necessary tables and relationships to meet the identified user and system requirements, such as the colonist,dependent,e-jet,trip,house,job,colonist-job and astronomer

- ❖ The logical design is normalized to Third Normal Form (3NF), Ensuring that data redundancy is minimized and data integrity is maintained, thus avoiding anomalies like update, insertion, and deletion anomalies.

- ❖ The use of primary and foreign keys in the design ensures that data is correctly linked between tables and that referential integrity is maintained.

In general, the given design meets the identified user and system requirements by providing an efficient and scalable method of storing, managing, and manipulating data related to E-Space Solutions.

2. Activity

Develop a relational database system according to the entity relational diagram create

SQL (Structured Query Language)

The language used to process and store information in relational databases is Structured Query Language or SQL. Data is grouped into rows and columns, which represent different data attributes, as well as relationships between values, in a relational database. SQL commands can be used to store, modify, delete, query and retrieve data from the database. To control and improve the performance of the database, SQL can also be used.

Importance if SQL (Structured Query Language)

Data management is built on Structured Query Language (SQL), which provides a standardised interface for interacting with databases. Its versatility and widespread use across industries make it a significant tool. SQL can be used to extract, manipulate, and organise large amounts of data, ensuring data integrity, facilitating database interaction, and allowing organisations to make educated decisions through data mining. Efficient the ability to handle data is a necessary talent for anyone dealing with data systems and information management, as it aids in software development, data analysis, and nearly every area of business operations.

Data types in my SQL

When creating tables, each columns maintain similar type of data such as numeric data values, text database so that they are a special keywords in my SQL to specify the data type of each column

I. Char (n)-fixed length characters (text)

If the data are with a fixed number of characters (A to Z, 0 to 9) and special character can be included

Eg: reg no char (4)

Tel char (10)

II. Variable length characters-varchar(n)

the number of the character expected are not fixed

Eg: studname varchar(15)

addr varchar(20)

III. int (n)-numbers (numerical value)

representing a numerical value as a whole number the maximum number of digits to be expected must be written within brackets

eg:maths mk int (3)

qty int (4)

IV. float (n,d) -decimal value

Representing a dotted decimal value by defining a complete number of digits expected and number of decimal places

eg : salary float (6,2)

avg float (4,1)

Insert year, month, date

Representing a date value with three section to represent the year(4numbers)month(2numbers) and the date(2numbers) while the separation can be / or _

eg : yyyy_mm_dd

yyyy/mm/dd

These SQL commands can be grouped into three types.

- **DDL – Data Definition Language**
- **DML – Data Manipulation Language**
- **DCL – Data Control Language**

Data Definition language (DDL)

DDL, or Data Definition Language, involves of SQL instructions that may be used to define the database structure. It only works with database schema descriptions and is used to construct and alter the structure of database objects in the database. DDL is a collection of SQL commands that are used to build, change, and erase database structures but not data.

The DDL operations CREATE, ALTER, DROP, and RENAME are frequently used in SQL queries

Typical DDL instructions include of:

CREATE - This SQL command allows you to create a new table. The user must enter information such as the table name, column names, and data types.

Create Database



A screenshot of the SQL Server Management Studio (SSMS) interface. The left pane shows the Object Explorer with a tree view of the database structure. The right pane contains a query window titled 'SQLQuery1.sql' with the following content:

```
create database e_spacesolution;
```

Below the query window, the 'Messages' pane displays the results of the execution:

```
Commands completed successfully.  
completion time: 2024-08-31T01:41:57.4763411+01:00
```

At the bottom of the screen, a status bar shows the connection details: 'DESKTOP-QDAGQF0\SQLEXPRESS' and 'DESKTOP-QDAGQF0\NewGen...'.

Figure 27 – Create database command

Create Table

The screenshot shows the Microsoft SQL Server Management Studio interface. In the Object Explorer, a database named 'e_spacesolution' is selected. In the center pane, a query window displays the following SQL code:

```
use e_spacesolution;
create table colonist(
    colonist_ID int primary key,
    first_name varchar(100)not null,
    last_name varchar(100)not null,
    sur_name varchar(100),
    dob date,
    qualification varchar(100),
    age date,
    gender varchar(100),
    number_of_members varchar(100));
```

Below the code, the 'Messages' section shows the output: "Commands completed successfully." and "Completion Time: 2024-09-01T04:53:02.1909907+01:00". At the bottom, a status bar indicates "Query executed successfully." and "Ready".

Figure 28 – Create table command

ALTER - This command allows you to insert, delete or modify columns in an existing table. The user should be familiar with the current table name and be able to add and delete it quickly.

ALTER STATEMENT

The screenshot shows the Microsoft SQL Server Management Studio interface. In the Object Explorer, a database named 'e_spacesolution' is selected. In the center pane, a query window displays the following SQL code:

```
select * from colonist;
```

Below the code, the 'Results' tab shows the output of the query:

colonist_ID	first_name	last_name	sur_name	dob	qualification	age	gender	number_of_members

At the bottom, a status bar indicates "Query executed successfully." and "Ready".

Figure 29 – alter table command

The screenshot shows the Microsoft SQL Server Management Studio interface. In the Object Explorer, a database named 'e_spacesolution' is selected. In the center pane, a query window titled 'SQLQuery1.sql - DESKTOP-Q0AQFDK\SQLEXPRESS.e_spacesolution (DESKTOP-Q0AQFDK\NewGen (54))' contains the following SQL code:

```
alter table colonist add civil_status varchar(100);
select * from colonist;
```

Below the code, the results pane shows a table with columns: colonist_ID, first_name, last_name, sur_name, dob, qualification, age, gender, number_of_members, and civil_status. The message bar at the bottom indicates 'Query executed successfully.'

Figure 30 – alter table command

DROP - This command deletes an existing table and its structure from the database.

DROP STATEMENT

The screenshot shows the Microsoft SQL Server Management Studio interface. In the Object Explorer, a database named 'e_spacesolution' is selected. In the center pane, a query window titled 'SQLQuery1.sql - DESKTOP-Q0AQFDK\SQLEXPRESS.e_spacesolution (DESKTOP-Q0AQFDK\NewGen (54))' contains the following SQL code:

```
drop table Job
```

The results pane shows a message: 'Commands completed successfully.' and 'Completion time: 2024-09-01T08:06:38.3250017+01:00'. The message bar at the bottom indicates 'Query executed successfully.'

Figure 31 – drop table command

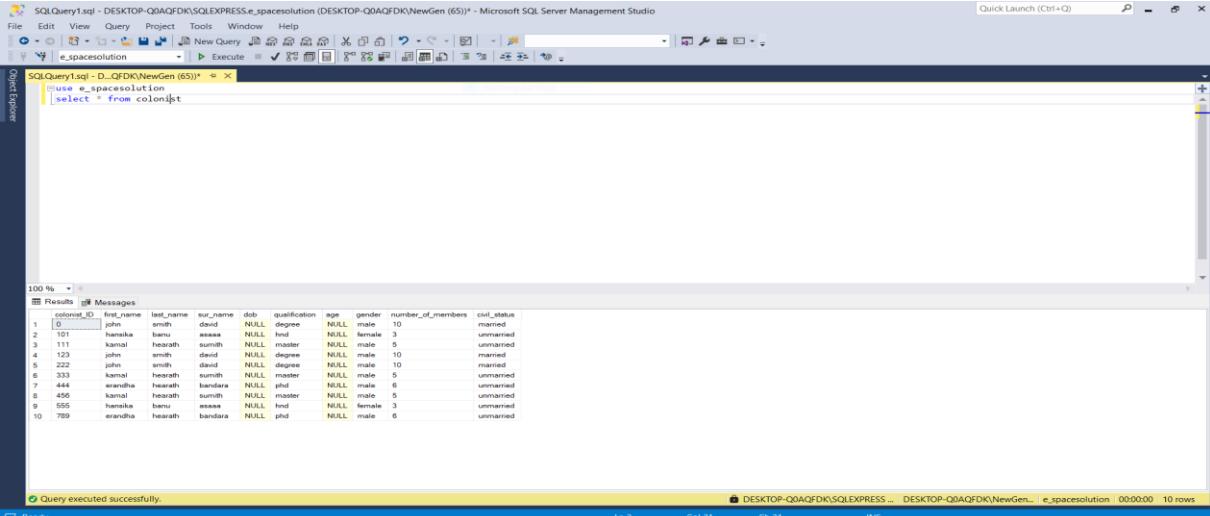
RENAME - This command renames an existing database item.

Data Manipulation Language (DML)

A subset of SQL (Structured Query Language) called Data Manipulation Language (DML) is used to manage data in databases. DML is concerned with accessing, adding, updating, and removing data inside tables, as opposed to Data Definition Language (DDL), which is focused on creating and changing the structure of the database.

Important DML instructions consist of:

SELECT - SELECT is the most important data manipulation operation in Structured Query Language. The SELECT command displays the records from the specified table. Furthermore, the WHERE clause is used to display a specific record from a certain column.



The screenshot shows the Microsoft SQL Server Management Studio interface. In the Object Explorer, a database named 'e_spacesolution' is selected. In the main query window, the following SQL code is written:

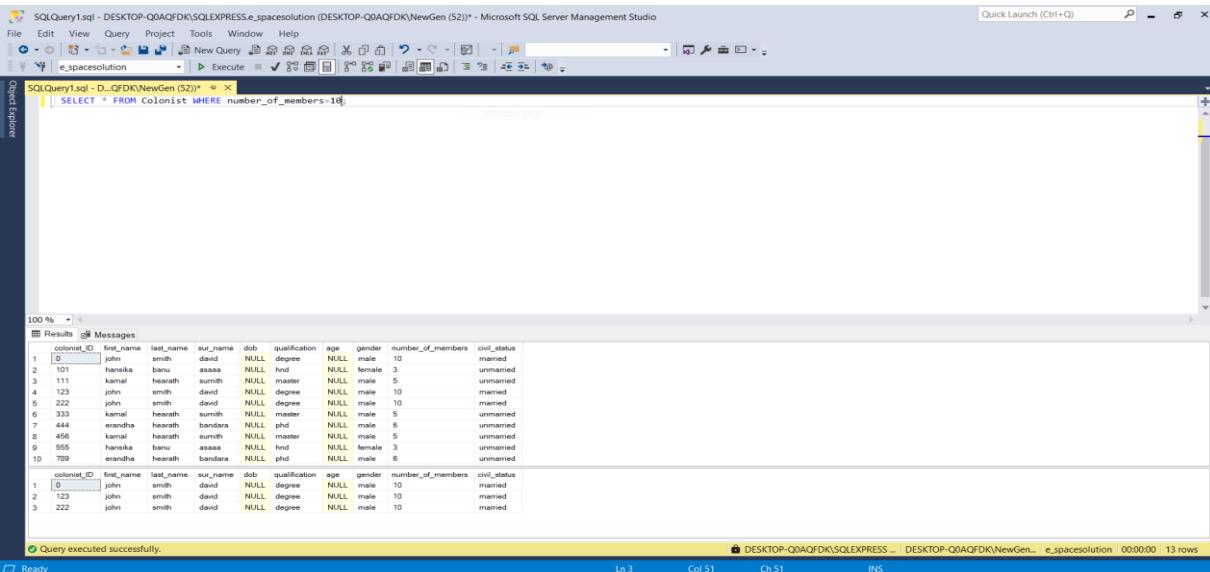
```
use e_spacesolution
select * from colonist
```

When executed, the results pane displays 10 rows of data from the 'colonist' table:

colonist_ID	first_name	last_name	sur_name	dob	qualification	age	gender	number_of_members	civil_status
1	john	smith	devid	NULL	degree	NULL	male	10	married
2	101	hansika	banu	asaaa	NULL	hnd	NULL	3	unmarried
3	111	kamal	hearth	sumith	NULL	master	NULL	5	unmarried
4	123	john	smith	devid	NULL	degree	NULL	10	married
5	222	john	smith	devid	NULL	degree	NULL	10	married
6	333	kamal	hearth	sumith	NULL	master	NULL	5	unmarried
7	444	erandha	hearth	bandara	NULL	phd	NULL	6	unmarried
8	456	kamal	hearth	sumith	NULL	master	NULL	5	unmarried
9	555	hansika	banu	asaaa	NULL	hnd	NULL	3	unmarried
10	789	erandha	hearth	bandara	NULL	phd	NULL	6	unmarried

At the bottom of the results pane, it says "Query executed successfully."

Figure 32 – select table command



The screenshot shows the Microsoft SQL Server Management Studio interface. In the Object Explorer, a database named 'e_spacesolution' is selected. In the main query window, the following SQL code is written:

```
use e_spacesolution
SELECT * FROM Colonist WHERE number_of_members=10;
```

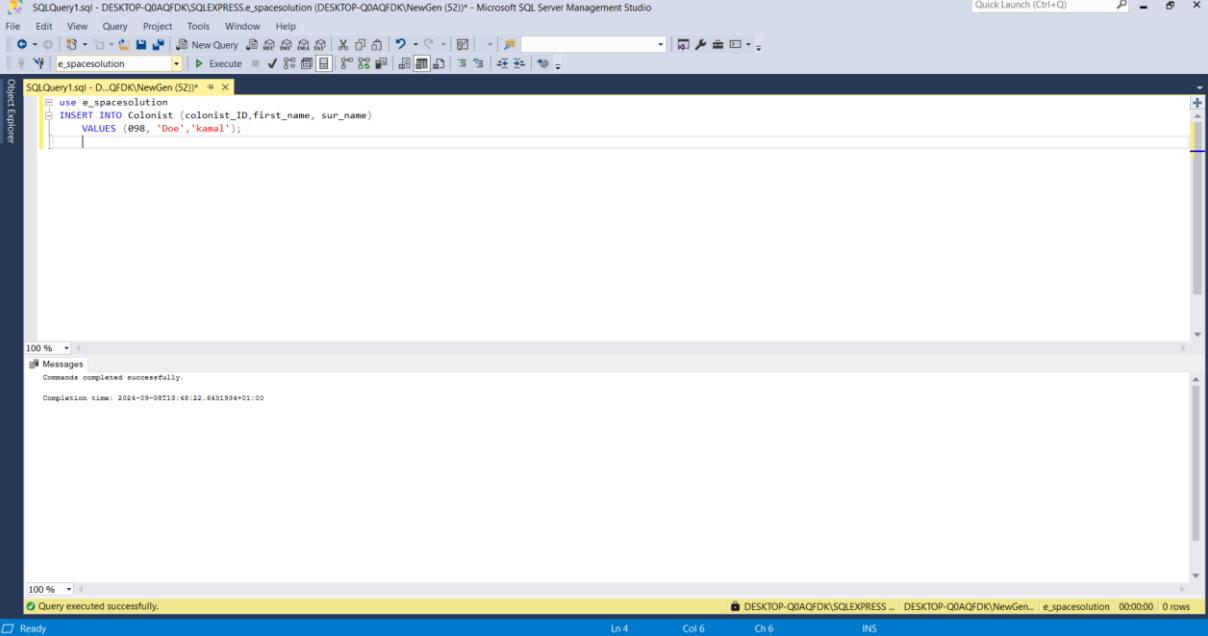
When executed, the results pane displays 3 rows of data from the 'colonist' table where the 'number_of_members' value is 10:

colonist_ID	first_name	last_name	sur_name	dob	qualification	age	gender	number_of_members	civil_status
1	john	smith	devid	NULL	degree	NULL	male	10	married
2	123	john	smith	devid	NULL	degree	NULL	10	married
3	222	john	smith	devid	NULL	degree	NULL	10	married

At the bottom of the results pane, it says "Query executed successfully."

Figure 33 –select table command

INSERT - Expands a table by adding information, records and additional rows or record.



The screenshot shows the Microsoft SQL Server Management Studio interface. A query window titled 'SQLQuery1.sql' is open, displaying the following SQL code:

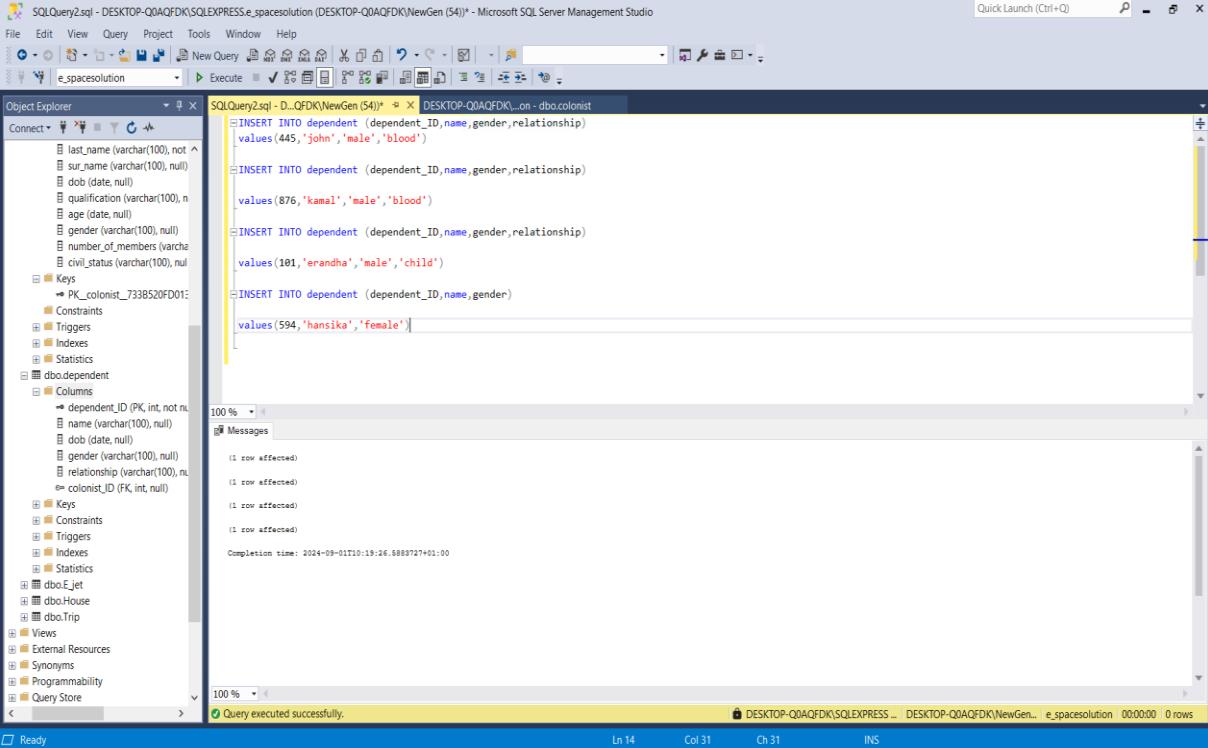
```
use e_spacesolution
INSERT INTO Colonist (colonist_ID, first_name, sur_name)
VALUES (998, 'Doe', 'kamal');
```

The 'Messages' pane at the bottom shows the output:

```
Commands completed successfully.
Completion time: 2024-09-08T13:48:32.6431934+01:00
```

The status bar at the bottom right indicates: DESKTOP-QQAQFDK\SQLEXPRESS - DESKTOP-QQAQFDK\NewGen... e_spacesolution 00:00:00 | 0 rows

Figure 34 –insert table command



The screenshot shows the Microsoft SQL Server Management Studio interface. A query window titled 'SQLQuery2.sql' is open, displaying the following SQL code:

```
SELECT * FROM dbo.colonist
SELECT * FROM dbo.dependent
```

The 'Messages' pane at the bottom shows the output:

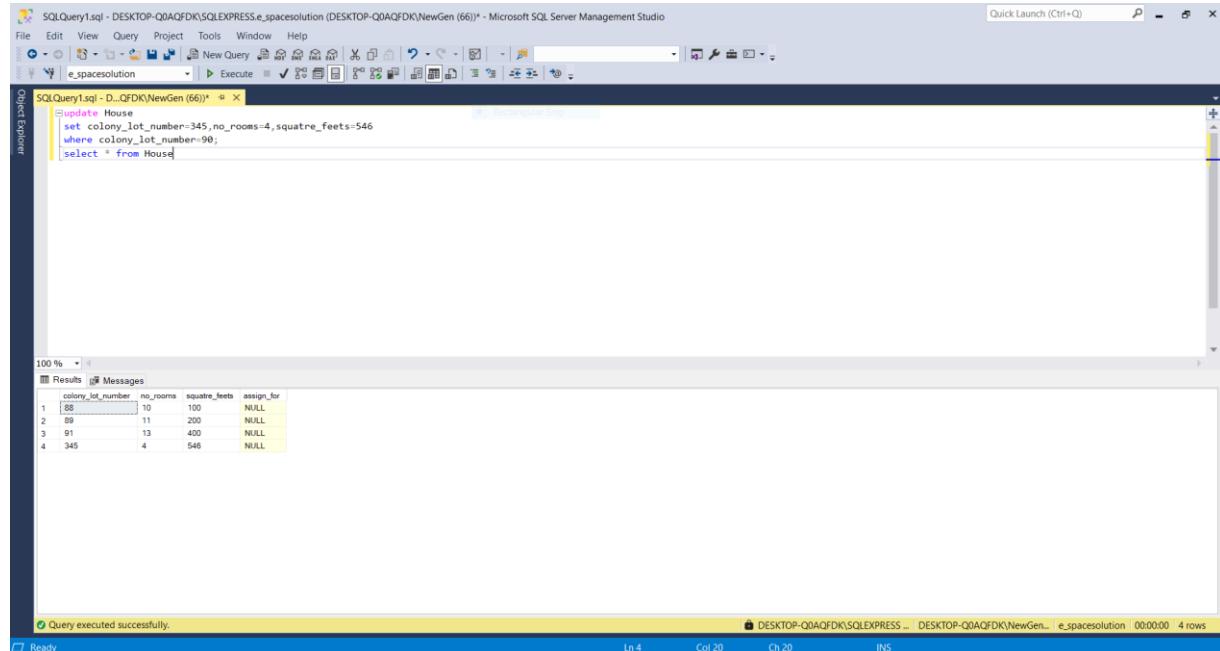
```
(1 row affected)
(1 row affected)
(1 row affected)
(1 row affected)

Completion time: 2024-09-01T10:19:26.5883727+01:00
```

The status bar at the bottom right indicates: DESKTOP-QQAQFDK\SQLEXPRESS - DESKTOP-QQAQFDK\NewGen... e_spacesolution 00:00:00 | 0 rows

Figure 35 –select table command

UPDATE - Adjusts current data in a table according to needed information by giving specific records.



The screenshot shows the Microsoft SQL Server Management Studio interface. In the Object Explorer, there is a connection to 'e_spacesolution'. In the center pane, a query window titled 'SQLQuery1.sql' contains the following SQL code:

```
update House
set colony_lot_number=345,no_rooms=4,squarefeet=546
where colony_lot_number=90;
select * from House
```

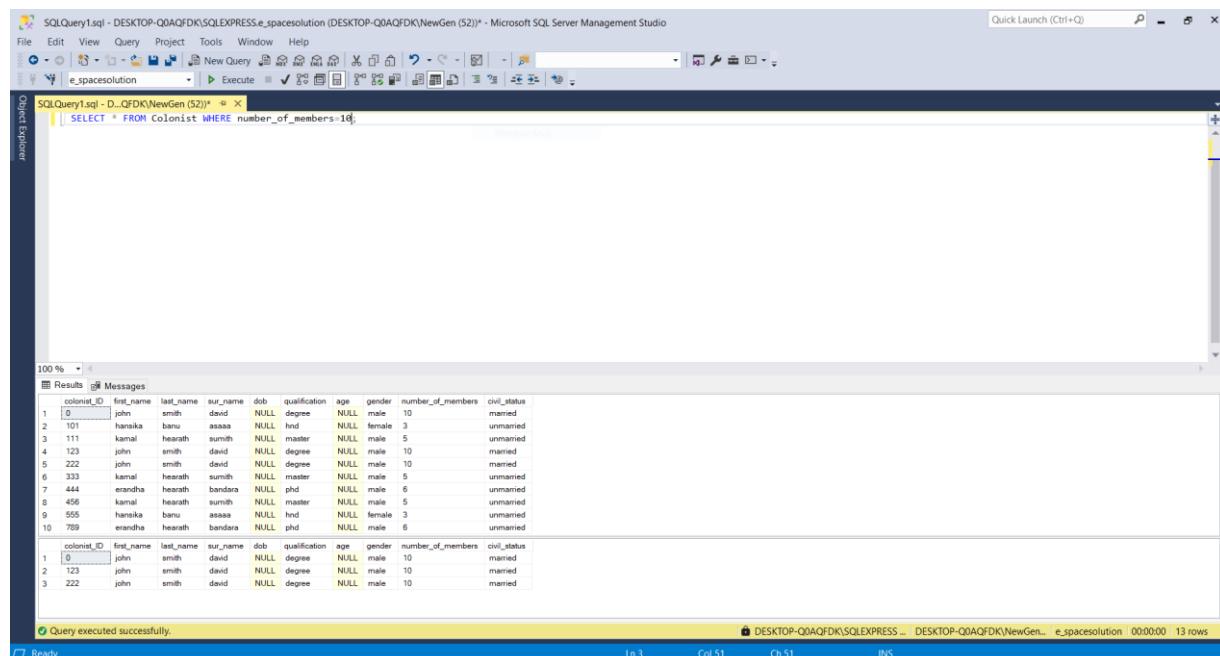
Below the code, the results pane shows a table with four rows of data:

colony_lot_number	no_rooms	squarefeet	assign_for
88	10	100	NULL
89	11	200	NULL
91	13	400	NULL
345	4	546	NULL

At the bottom of the interface, a message bar indicates: 'Query executed successfully.' and 'Ln 4 Col 20 Ch 20 INS'.

Figure 36 –update table command

WHERE - The WHERE clause of a SELECT query specifies the search parameters for determining which rows are returned. Search criteria are also known as predicates.



The screenshot shows the Microsoft SQL Server Management Studio interface. In the Object Explorer, there is a connection to 'e_spacesolution'. In the center pane, a query window titled 'SQLQuery1.sql' contains the following SQL code:

```
SELECT * FROM Colonist WHERE number_of_members>10;
```

Below the code, the results pane shows two tables. The first table is the 'Colonist' table with 10 rows of data:

colonist_ID	first_name	last_name	sur_name	dob	qualification	age	gender	number_of_members	civil_status	
1	john	smith	david	NULL	degree	NULL	male	10	married	
2	101	hansika	banu	aaaaa	NULL	hnd	NULL	female	3	unmarried
3	111	kamal	hearth	sunith	NULL	master	NULL	male	5	unmarried
4	123	john	smith	david	NULL	degree	NULL	male	10	married
5	222	john	smith	david	NULL	degree	NULL	male	10	married
6	333	kamal	hearth	sunith	NULL	master	NULL	male	5	unmarried
7	444	erandha	hearth	bandara	NULL	phd	NULL	male	6	unmarried
8	456	kamal	hearth	sunith	NULL	master	NULL	male	5	unmarried
9	555	hansika	banu	aaaaa	NULL	hnd	NULL	female	3	unmarried
10	789	erandha	hearth	bandara	NULL	phd	NULL	male	6	unmarried

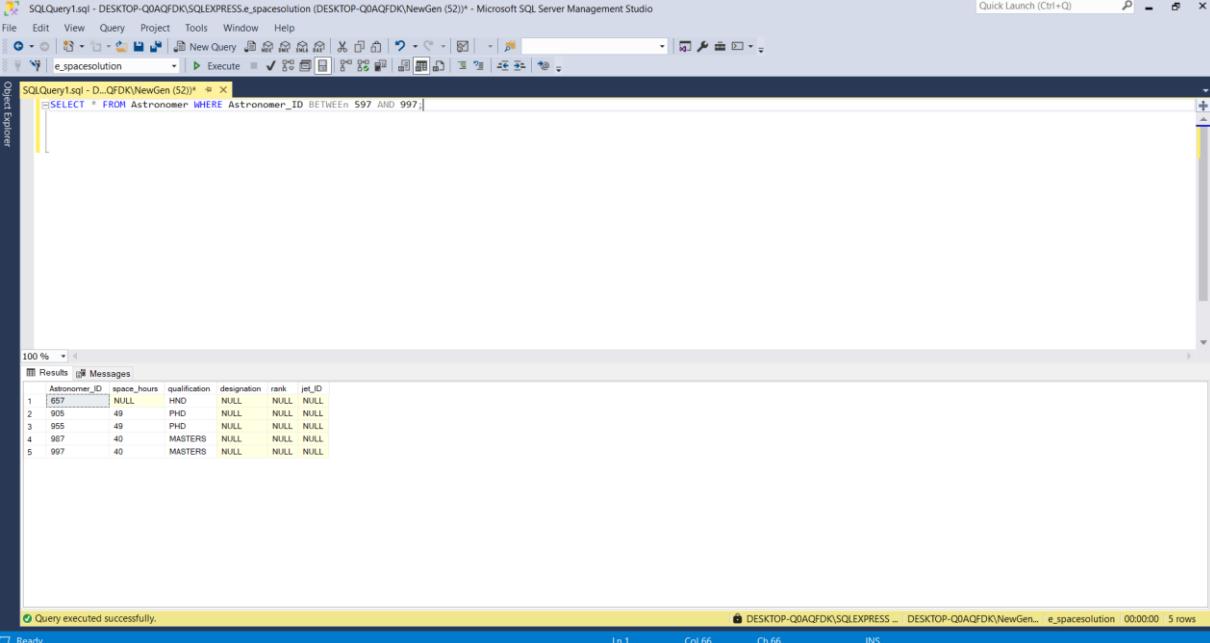
The second table is a subset of the first, containing only rows where 'number_of_members' is greater than 10:

colonist_ID	first_name	last_name	sur_name	dob	qualification	age	gender	number_of_members	civil_status	
1	john	smith	david	NULL	degree	NULL	male	10	married	
2	123	john	smith	david	NULL	degree	NULL	male	10	married
3	222	john	smith	david	NULL	degree	NULL	male	10	married

At the bottom of the interface, a message bar indicates: 'Query executed successfully.' and 'Ln 3 Col 51 Ch 51 INS'.

Figure 37 –where command

BETWEEN - Using the SQL BETWEEN condition (inclusive), you may rapidly check whether an expression falls inside a certain range of values. Text, dates, and numbers are all valid values. It applies to statements that SELECT, INSERT, UPDATE, or DELETE data. The SQL BETWEEN Condition will return entries that fall between value1 and value2.



The screenshot shows the Microsoft SQL Server Management Studio interface. A query window is open with the following SQL code:

```
SQLQuery1.sql - DESKTOP-Q0AQFDK\SQLEXPRESS.e_spacesolution (DESKTOP-Q0AQFDK\NewGen (52)) - Microsoft SQL Server Management Studio
File Edit View Query Project Tools Window Help
Object Explorer
SQLQuery1.sql - D:\QFDK\NewGen (52)*
SELECT * FROM Astronomer WHERE Astronomer_ID BETWEEN 597 AND 997;
```

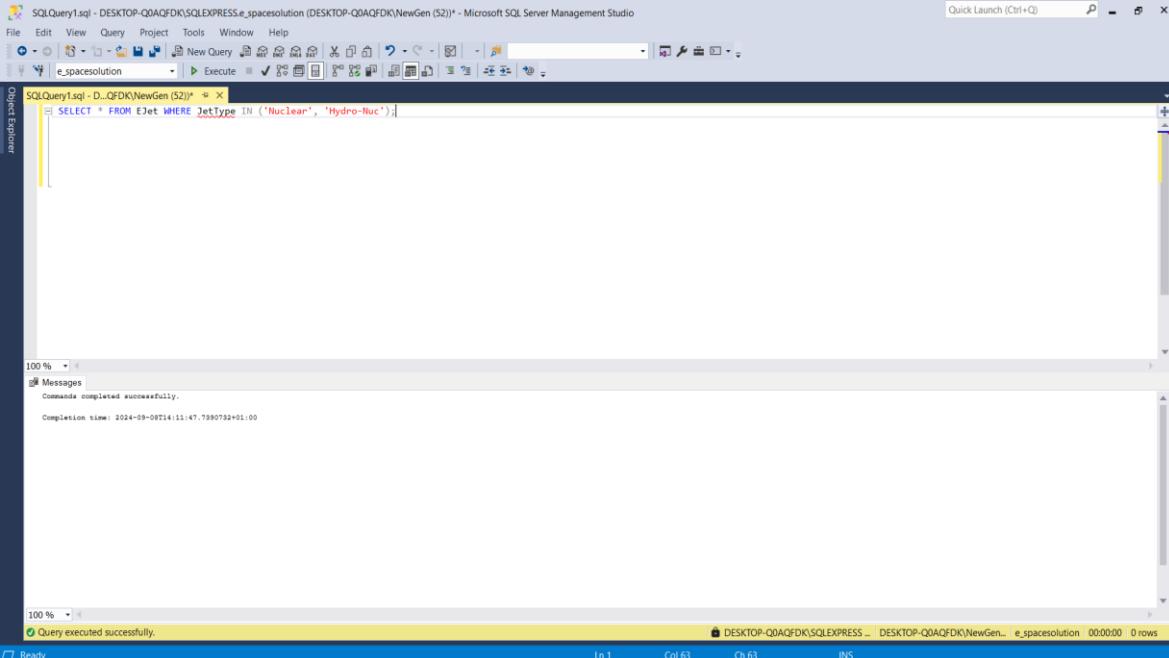
The results pane displays the following data:

Astronomer_ID	Space_Hours	qualification	designation	rank	jetID
1	657	NULL	HND	NULL	NULL
2	905	49	PHD	NULL	NULL
3	955	49	PHD	NULL	NULL
4	987	40	MASTERS	NULL	NULL
5	997	40	MASTERS	NULL	NULL

Below the results, a message bar indicates: "Query executed successfully." and "Ready".

Figure 38–between condition

IN - The SQL IN condition (also known as the IN operator) allows you to rapidly check if an expression matches any value in a list of values. It can help avoid the need for several OR criteria when used with a SELECT, INSERT, UPDATE, or DELETE query.



The screenshot shows the Microsoft SQL Server Management Studio interface. A query window is open with the following SQL code:

```
SQLQuery1.sql - DESKTOP-Q0AQFDK\SQLEXPRESS.e_spacesolution (DESKTOP-Q0AQFDK\NewGen (52)) - Microsoft SQL Server Management Studio
File Edit View Query Project Tools Window Help
Object Explorer
SQLQuery1.sql - D:\QFDK\NewGen (52)*
SELECT * FROM Ejet WHERE JetType IN ('Nuclear', 'Hydro-Nuc');
```

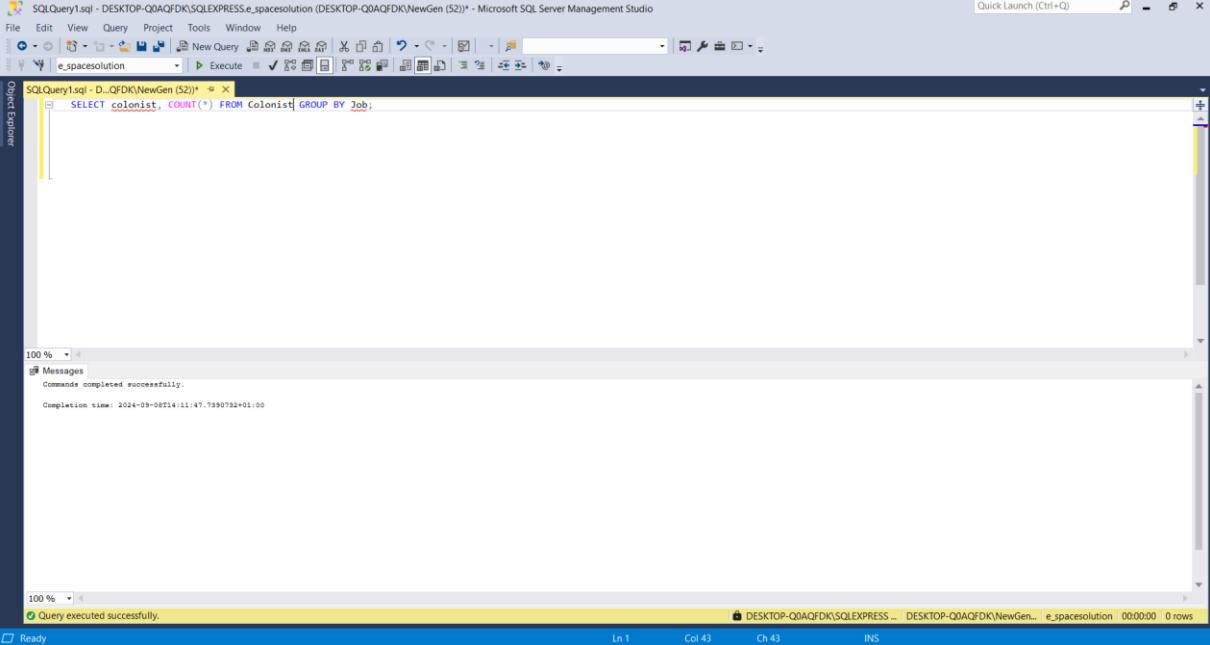
The results pane displays the following data:

Ejet	JetType
1	Nuclear
2	Hydro-Nuc

Below the results, a message bar indicates: "Commands completed successfully." and "Compilation time: 2024-09-08T14:11:47.7980732+01:00".

Figure 39– In command

GROUP BY - Finding summary rows, such as "Find the number of colonist in, E_space solution" is accomplished with the GROUP BY Command, which groups rows with comparable values.



```
SQLQuery1.sql - DESKTOP-QOAQFDK\SQLEXPRESS.e_spacesolution (DESKTOP-QOAQFDK\NewGen (S2)) - Microsoft SQL Server Management Studio
File Edit View Query Project Tools Window Help
New Query Execute
e_spacesolution Object Explorer
SQLQuery1.sql - DESKTOP-QOAQFDK\NewGen (S2)*
SELECT colonist, COUNT(*) FROM Colonist GROUP BY Job;
```

Messages

Commands completed successfully.

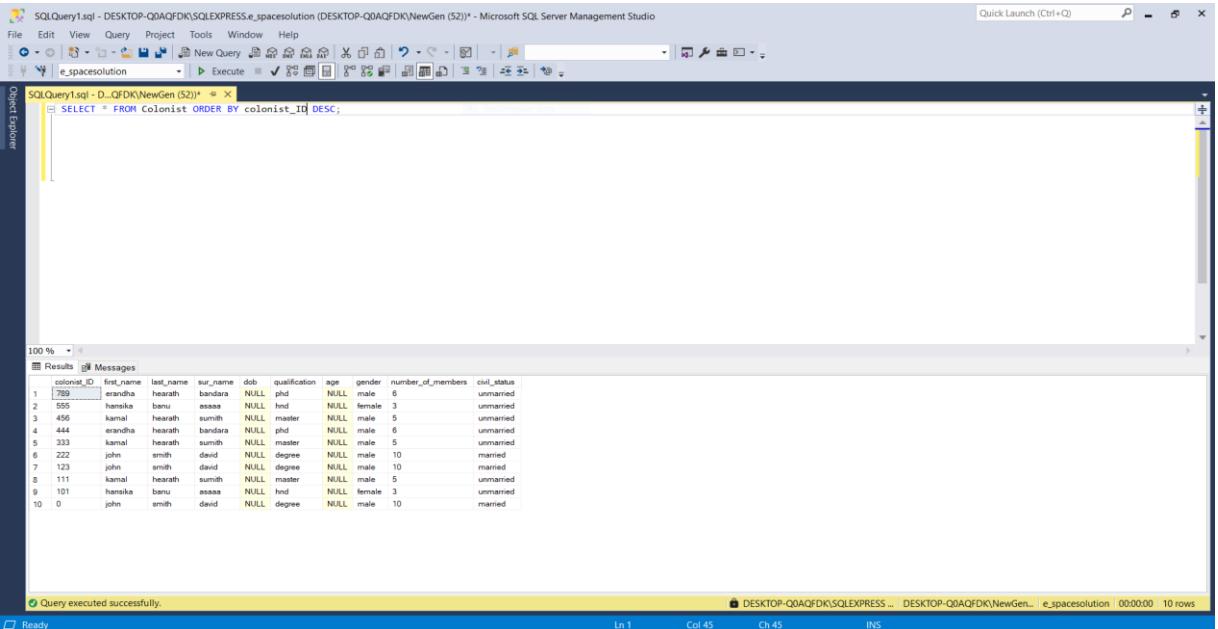
Completion time: 2024-09-08T14:11:47.7390732+01:00

Query executed successfully.

Ready Col 43 Ch 43 INS

Figure 40 – Group by command

ORDER BY - If no subsequent requests are made, the database server returns the rows of a table in random order. It is frequently beneficial to examine a table's rows in a more meaningful order.



```
SQLQuery1.sql - DESKTOP-QOAQFDK\SQLEXPRESS.e_spacesolution (DESKTOP-QOAQFDK\NewGen (S2)) - Microsoft SQL Server Management Studio
File Edit View Query Project Tools Window Help
New Query Execute
e_spacesolution Object Explorer
SQLQuery1.sql - DESKTOP-QOAQFDK\NewGen (S2)*
SELECT * FROM Colonist ORDER BY colonist_ID DESC;
```

Results

colonist_ID	first_name	last_name	sur_name	dob	qualification	age	gender	number_of_members	civil_status	
1	eranda	hearth	bandara	NULL	phd	NULL	male	6	unmarried	
2	eranda	hearth	benu	NULL	assess	NULL	female	3	unmarried	
3	kamal	hearth	sunith	NULL	master	NULL	male	5	unmarried	
4	eranda	hearth	bandara	NULL	phd	NULL	male	6	unmarried	
5	333	kamal	hearth	sunith	NULL	master	NULL	male	5	unmarried
6	222	john	smith	deevi	NULL	degree	NULL	male	10	married
7	123	john	smith	david	NULL	degree	NULL	male	10	married
8	111	kamal	hearth	sunith	NULL	master	NULL	male	5	unmarried
9	101	hansika	banu	assaaa	NULL	hd	NULL	female	3	unmarried
10	0	john	smith	david	NULL	degree	NULL	male	10	married

Messages

Query executed successfully.

Ready Col 45 Ch 45 INS

Figure 41 – Order by command

Having

The SELECT statement's HAVING clause inserts the condition in the groups specified by the GROUP BY clause. The 'GROUP BY' clause of the 'SELECT' query occurs before this SQL clause. Because the WHERE clause does not operate with SQL's aggregate functions, this clause is used instead. The records in SQL queries are filtered using the WHERE and HAVING clauses.

Data Control Language (DCL)

A subset of SQL (Structured Query Language) called Data Control Language (DCL) is used to manage rights and access privileges inside databases. DCL commands manage who can do what inside the database environment, including what operations and activities may be carried out, and they also regulate access to the database and its objects.

In DCL, the two main instructions are:

GRANT - This command enables a user or group of users to carry out particular operations on database objects by granting them specific capabilities. For example, giving a user access to SELECT, INSERT, UPDATE, and DELETE data in a table

REVOKE - This function is used to remove or revoke rights that have already been given to users or roles. Certain rights that were previously given with the GRANT command are removed with this command.

The following are some of the most regularly used SQL constraints.

SQL Constraints

SQL constraints are directives or conditions applied to table columns in order to preserve consistency and data integrity in a database. They specify the restrictions and requirements that information stored in a table must follow in order to ensure the data's validity, accuracy, and reliability.

Foreign key constraint: Establishing a link between two tables and ensuring referential integrity is the purpose of the foreign key constraint. In order to guarantee that values in the foreign key column exist as values in the referenced primary key column, the foreign key in one table refers to the main key in another table.

```

create table dependent(
dependent_ID int primary key,
name varchar(100),
dob date,
gender varchar(100),
relationship varchar(100),
colonist_ID int FOREIGN KEY references colonist(colonist_ID));

```

Figure 42 –foreign key constraint

Primary key constraint: ensures that every entry in a table is uniquely identified. It guarantees that no values are null may exist in the column or set of columns designated as the primary key, and that all values must be unique to specify that records uniquely.

```

create table colonist(
colonist_ID int primary key,
first_name varchar(100)not null,
last_name varchar(100)not null,
sur_name varchar(100),
dob date,
qualification varchar(100),
age date,
gender varchar(100),
number_of_members varchar(100));

```

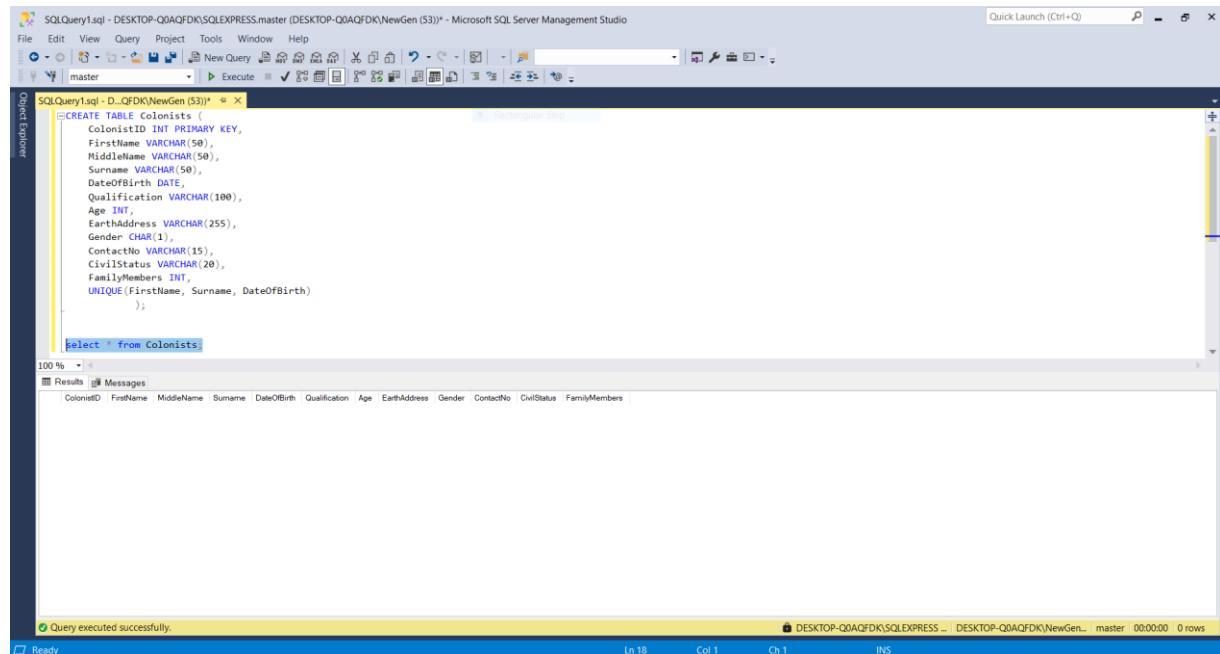
Figure 43 –primary key constraint

Not Null Constraint: Ensures that a column cannot have a NULL value.

```
create table colonist(
colonist_ID int primary key,
first_name varchar(100)not null,
last_name varchar(100)not null,
sur_name varchar(100),
dob date,
qualification varchar(100),
age date,
gender varchar(100),
number_of_members varchar(100));
```

Figure 44 – not null constraint

Unique Constraint: Allows null values yet functions similarly to a main key. It guarantees column or group of columns in order to maintain uniqueness of each value.



The screenshot shows the Microsoft SQL Server Management Studio interface. In the Object Explorer, a table named 'Colonists' is expanded, showing its schema. The 'ColonistID' column is defined as INT PRIMARY KEY. Other columns include FirstName, MiddleName, Surname, DateOfBirth, Qualification, Age, EarthAddress, Gender, ContactNo, CivilStatus, and FamilyMembers. A UNIQUE constraint is defined on the combination of FirstName, Surname, and DateOfBirth. Below the schema, a 'Select * from Colonists' query is run, and the results pane shows no rows returned. At the bottom, a message indicates the query was executed successfully.

```
CREATE TABLE Colonists (
    ColonistID INT PRIMARY KEY,
    FirstName VARCHAR(50),
    MiddleName VARCHAR(50),
    Surname VARCHAR(50),
    DateOfBirth DATE,
    Qualification VARCHAR(100),
    Age INT,
    EarthAddress VARCHAR(255),
    Gender CHAR(1),
    ContactNo VARCHAR(15),
    CivilStatus VARCHAR(20),
    FamilyMembers INT,
    UNIQUE (FirstName, Surname, DateOfBirth)
);
```

```
Select * from Colonists;
```

Query executed successfully.

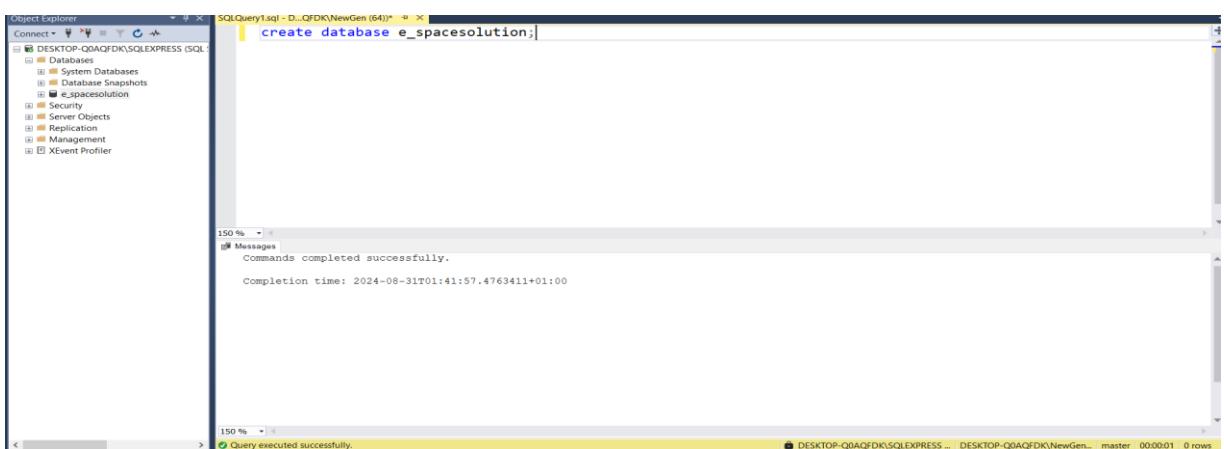
Figure 45 – unique constraint

Creating the relational database system according to the ERD

I'm using Microsoft SQL 2022 to construct a relational database that match the provided ER Diagram. Using the DDL (Data Definition Language), which has commands like CREATE, ALTER, USE, and DROP, I am beginning to construct the database.

I will build a database for E-SPACE SOLUTION. To do this, I will input the command "CREATE DATABASE" along with the name of the database I intend to utilize.

CREATE database



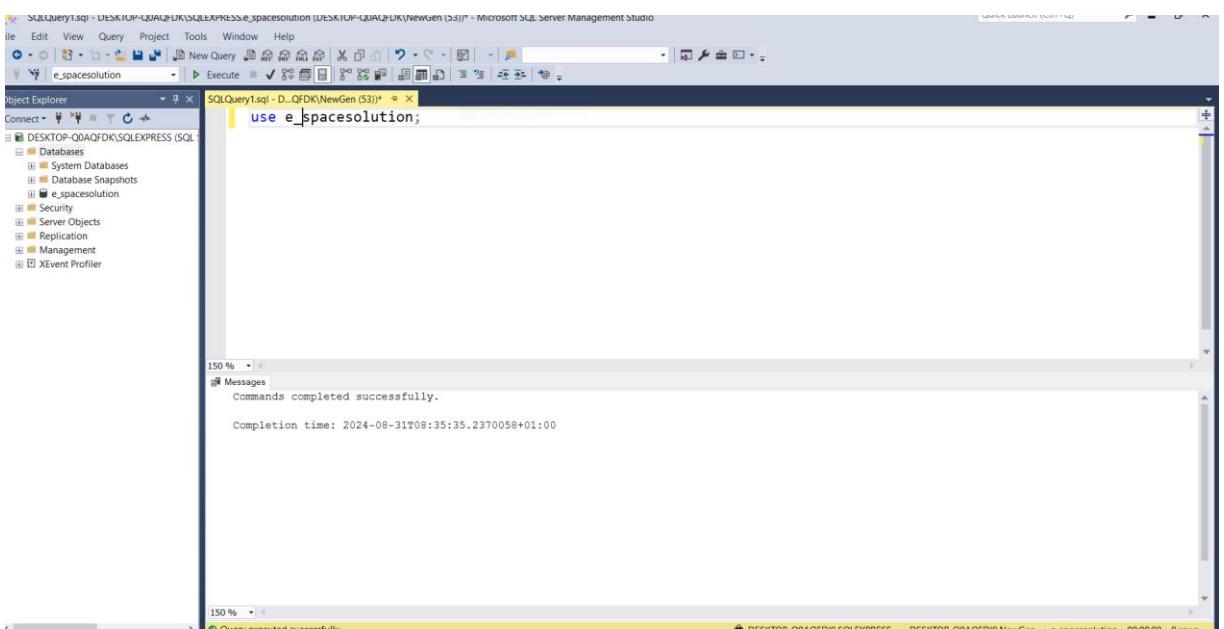
A screenshot of Microsoft SQL Server Management Studio (SSMS) showing the creation of a database. The Object Explorer on the left shows the database 'e_spacesolution' under 'DESKTOP-Q0AQFDK\SQLEXPRESS (SQL Server)'. A query window titled 'SQLQuery1.sql - D:\QFDK\NewGen (64)*' contains the command: 'create database e_spacesolution;'. The Messages pane below the query window displays the output: 'Commands completed successfully.' and 'Completion time: 2024-08-31T01:41:57.4763411+01:00'. The status bar at the bottom indicates 'Query executed successfully.'

```
create database e_spacesolution;
```

Commands completed successfully.
Completion time: 2024-08-31T01:41:57.4763411+01:00
Query executed successfully.

Figure 46– create database command

USE statement



A screenshot of Microsoft SQL Server Management Studio (SSMS) showing the use of a database. The Object Explorer on the left shows the database 'e_spacesolution' under 'DESKTOP-Q0AQFDK\SQLEXPRESS (SQL Server)'. A query window titled 'SQLQuery1.sql - D:\QFDK\NewGen (53)*' contains the command: 'use e_spacesolution;'. The Messages pane below the query window displays the output: 'Commands completed successfully.' and 'Completion time: 2024-08-31T08:35:35.2370058+01:00'. The status bar at the bottom indicates 'Query executed successfully.'

```
use e_spacesolution;
```

Commands completed successfully.
Completion time: 2024-08-31T08:35:35.2370058+01:00
Query executed successfully.

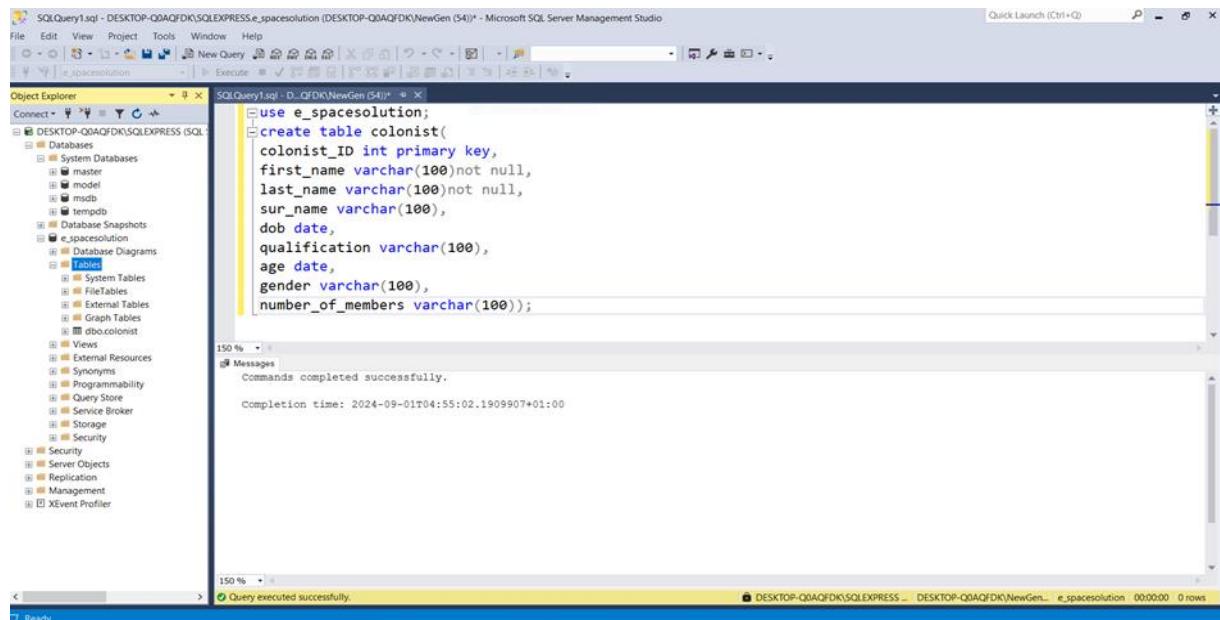
Figure 47– use database command

I have inserted a command **create database E-SPACE SOLUTION** and then I executed it. We need access the created database, for that we must insert **use** command.

Now we are using **E_space solution DATABASE**.

We need to **START CREATING TABLES**. To create the tables for the database system, we need to specify the **PRIMARY KEY**, **FOREIGN KEY REFERENCES** and the **NECESSARY CONSTRAINTS**.

CREATE colonist table



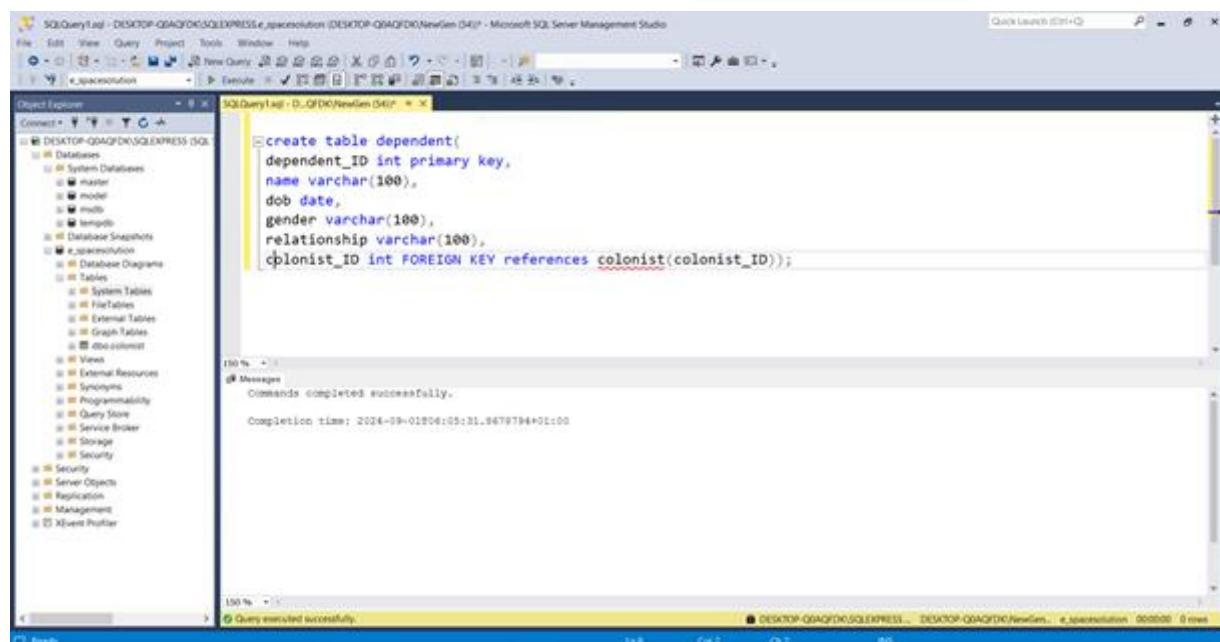
The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left shows a connection to 'DESKTOP-QQAQFDK\SQLEXPRESS_e_spacesolution'. The 'Tables' node under the database is expanded, showing the 'colonist' table. The 'Script' button next to the table name is highlighted. The central pane displays the T-SQL code for creating the 'colonist' table:

```
use e_spacesolution;
create table colonist(
    colonist_ID int primary key,
    first_name varchar(100)not null,
    last_name varchar(100)not null,
    sur_name varchar(100),
    dob date,
    qualification varchar(100),
    age date,
    gender varchar(100),
    number_of_members varchar(100));
```

The status bar at the bottom indicates 'Query executed successfully.'

Figure 48– create table colonist

CREATE dependent table



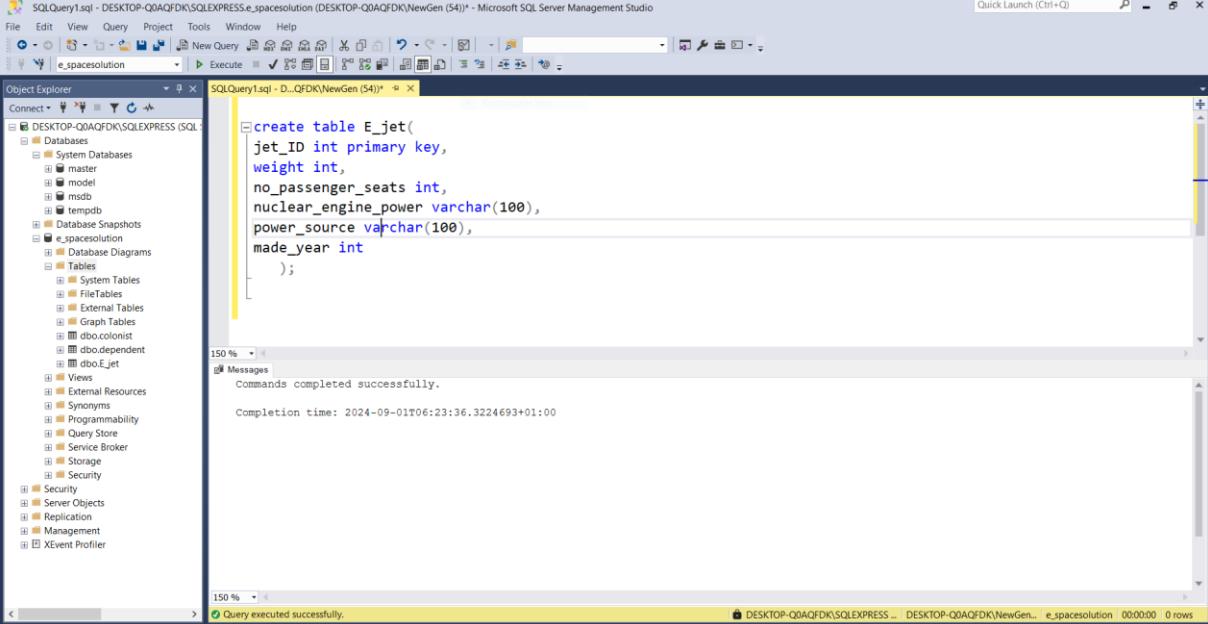
The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left shows a connection to 'DESKTOP-QQAQFDK\SQLEXPRESS_e_spacesolution'. The 'Tables' node under the database is expanded, showing the 'dependent' table. The 'Script' button next to the table name is highlighted. The central pane displays the T-SQL code for creating the 'dependent' table:

```
create table dependent(
    dependent_ID int primary key,
    name varchar(100),
    dob date,
    gender varchar(100),
    relationship varchar(100),
    colonist_ID int FOREIGN KEY references colonist(colonist_ID));
```

The status bar at the bottom indicates 'Query executed successfully.'

Figure 49– create table dependent

CREATE E_jet table



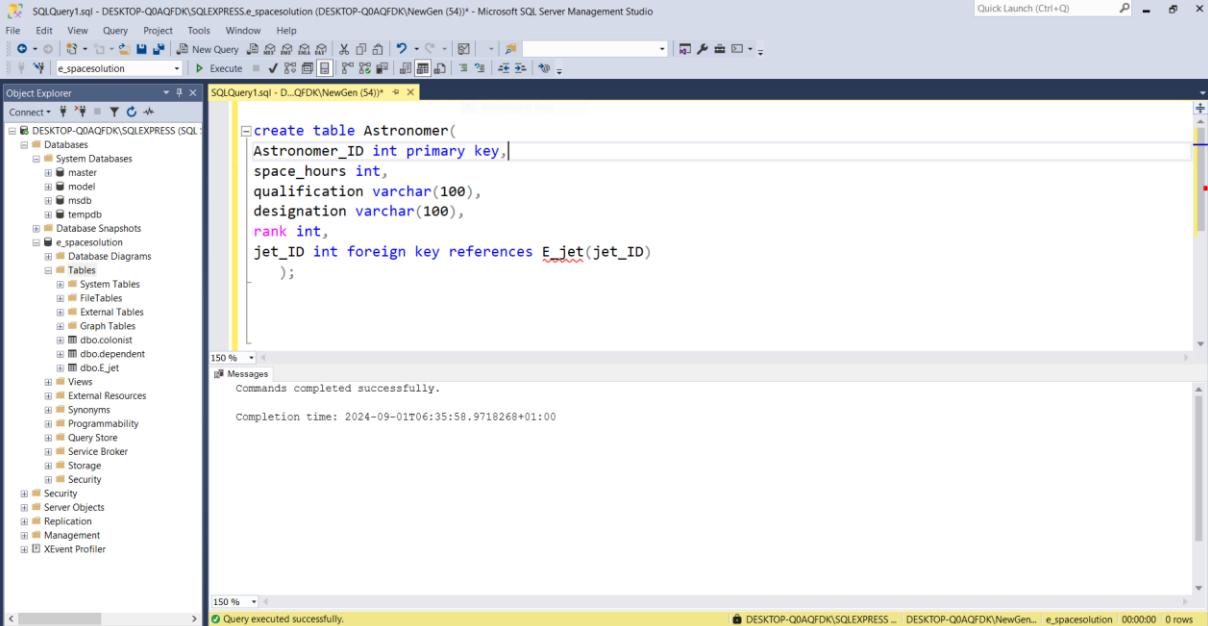
The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left shows the database structure, including the e_spacesolution database and its tables. The central pane displays the following SQL code:

```
create table E_jet(
jet_ID int primary key,
weight int,
no_passenger_seats int,
nuclear_engine_power varchar(100),
power_source varchar(100),
made_year int
);
```

The status bar at the bottom indicates "Query executed successfully." and "Completion time: 2024-09-01T06:23:36.3224693+01:00".

Figure 50– create table E_jet

CREATE astronomer table



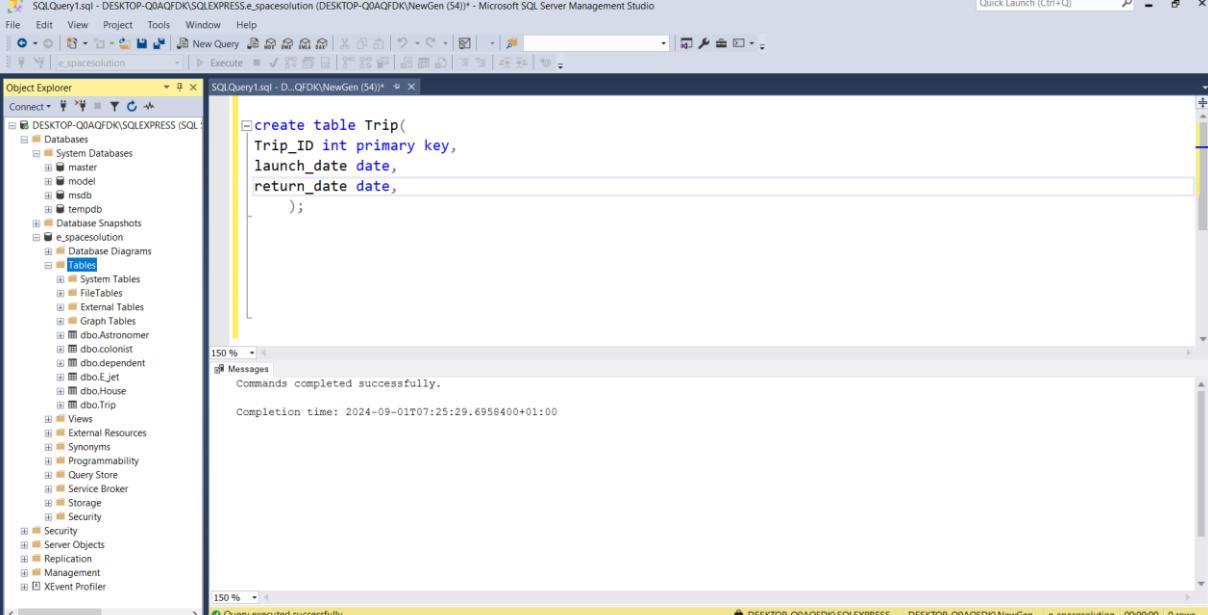
The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left shows the database structure, including the e_spacesolution database and its tables. The central pane displays the following SQL code:

```
create table Astronomer(
Astronomer_ID int primary key,
space_hours int,
qualification varchar(100),
designation varchar(100),
rank int,
jet_ID int foreign key references E_jet(jet_ID)
);
```

The status bar at the bottom indicates "Query executed successfully." and "Completion time: 2024-09-01T06:35:58.9718268+01:00".

Figure 51– create table astronomer

CREATE trip table



The screenshot shows the Microsoft SQL Server Management Studio interface. In the Object Explorer, the database 'e_spacesolution' is selected. In the center pane, a T-SQL script is being run:

```
create table Trip(
    Trip_ID int primary key,
    launch_date date,
    return_date date,
);
```

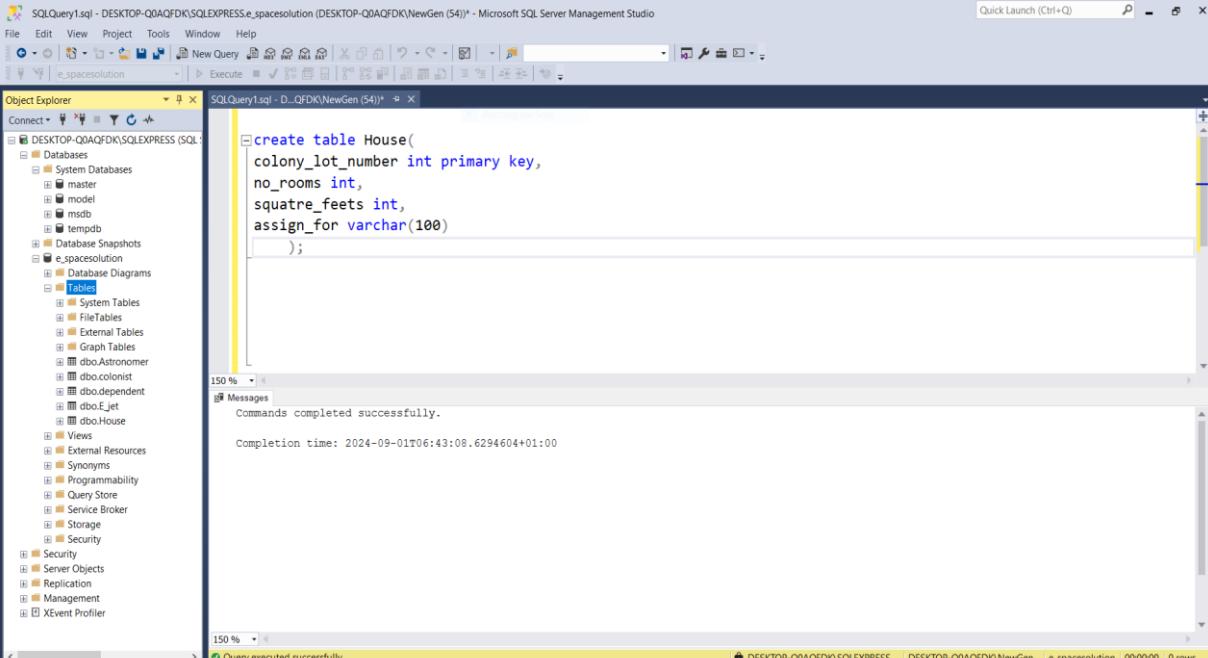
The 'Messages' pane at the bottom displays the results of the query execution:

Commands completed successfully.
Completion time: 2024-09-01T07:25:29.6958400+01:00

Query executed successfully.

Figure 52– create table trip

CREATE house table



The screenshot shows the Microsoft SQL Server Management Studio interface. In the Object Explorer, the database 'e_spacesolution' is selected. In the center pane, a T-SQL script is being run:

```
create table House(
    colony_lot_number int primary key,
    no_rooms int,
    square_feet int,
    assign_for varchar(100),
);
```

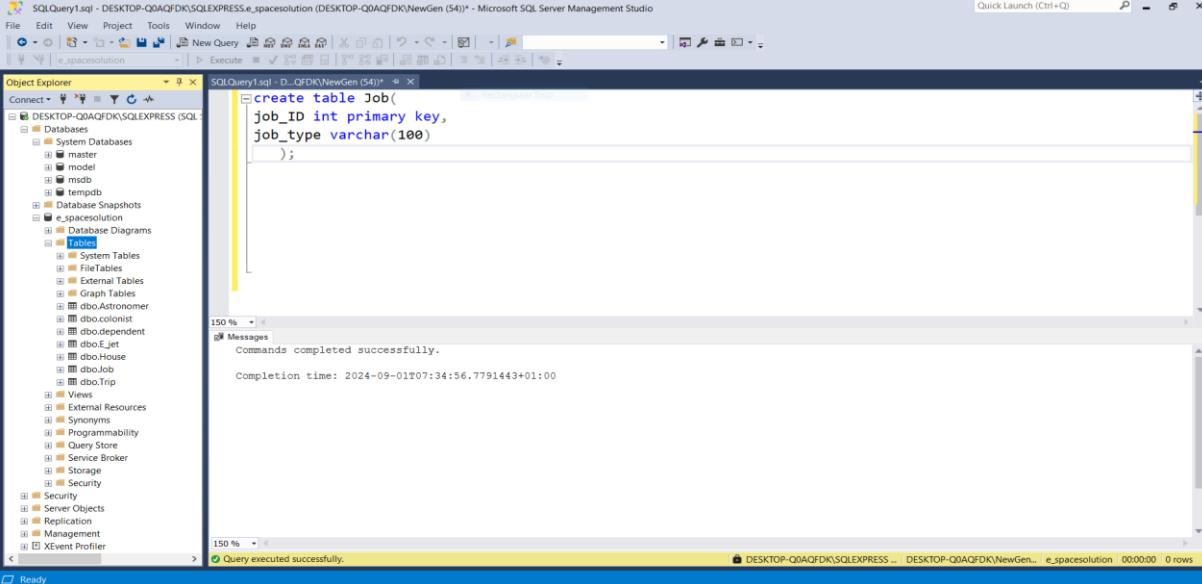
The 'Messages' pane at the bottom displays the results of the query execution:

Commands completed successfully.
Completion time: 2024-09-01T06:43:08.6294604+01:00

Query executed successfully.

Figure 53– create table house

CREATE job table



The screenshot shows the Microsoft SQL Server Management Studio interface. In the Object Explorer on the left, the database 'e_spacesolution' is selected, and the 'Tables' node is expanded. In the center pane, a T-SQL script is being run:

```
create table Job(
    job_ID int primary key,
    job_type varchar(100)
);
```

The status bar at the bottom indicates 'Query executed successfully.'

Figure 54– create table job

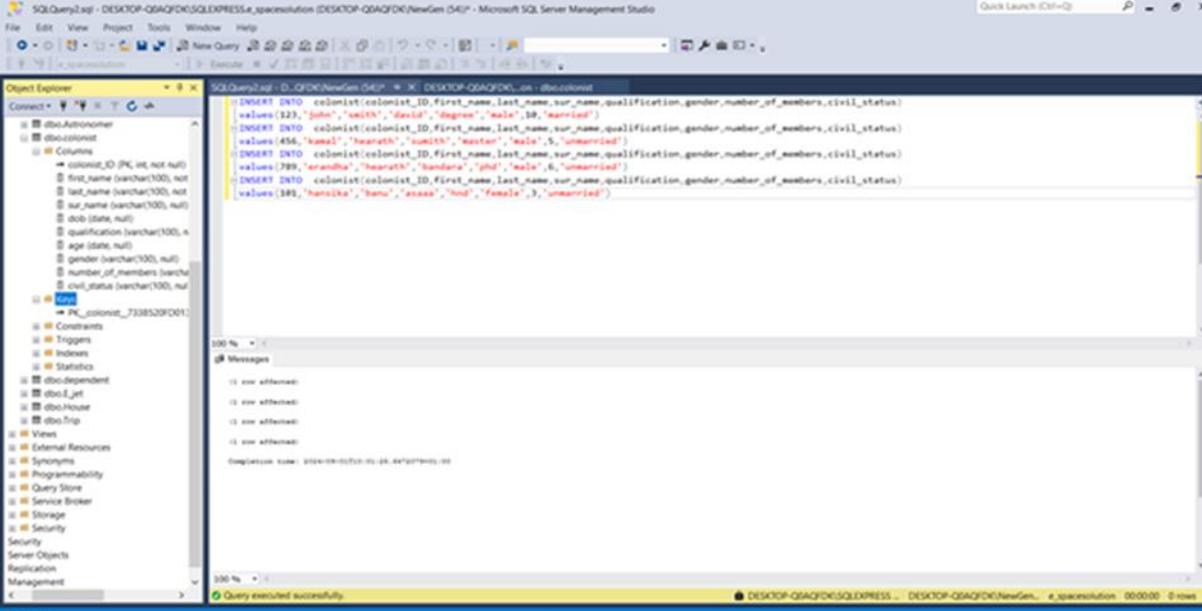
Insert data in to the table are 2 method

1st method

As we have created the table, now we must insert data into it. In order to insert data, we us the list of DML (Data Manipulation Language) command which is Insert into, Update and Select.

First, I'm going to insert data , in 1st methods . Using INSERT INTO command.

Insert data into colonist table



The screenshot shows the Microsoft SQL Server Management Studio interface. In the Object Explorer on the left, the database 'e_spacesolution' is selected, and the 'Tables' node is expanded. In the center pane, a T-SQL script is being run:

```
INSERT INTO colonist(colonist_ID, first_name, last_name, sur_name, qualification, gender, number_of_members, civil_status)
values('123', 'john', 'smith', 'doe', 'male', '18', 'married');
INSERT INTO colonist(colonist_ID, first_name, last_name, sur_name, qualification, gender, number_of_members, civil_status)
values('456', 'kumar', 'hearth', 'sumit', 'master', 'male', '5', 'unmarried');
INSERT INTO colonist(colonist_ID, first_name, last_name, sur_name, qualification, gender, number_of_members, civil_status)
values('789', 'erandha', 'hearth', 'bandara', 'phd', 'male', '6', 'unmarried');
INSERT INTO colonist(colonist_ID, first_name, last_name, sur_name, qualification, gender, number_of_members, civil_status)
values('389', 'hanika', 'benu', 'aaaa', 'Female', '3', 'unmarried');
```

The status bar at the bottom indicates 'Query executed successfully.'

Figure 55– create table colonist

The screenshot shows the Microsoft SQL Server Management Studio interface. In the Object Explorer, the 'colonist' table is selected under the 'dbo.dependent' schema. A query window titled 'SQLQuery2.sql' contains the following T-SQL code:

```

-- Create table colonist
CREATE TABLE colonist(
    colonist_ID INT IDENTITY(1,1) PRIMARY KEY,
    first_name VARCHAR(100),
    last_name VARCHAR(100),
    sur_name VARCHAR(100),
    dob DATE,
    qualification VARCHAR(100),
    age INT,
    gender VARCHAR(100),
    number_of_members INT,
    civil_status VARCHAR(100)
);

-- Insert data into colonist
INSERT INTO colonist(first_name, last_name, sur_name, qualification, gender, number_of_members, civil_status)
VALUES('john', 'smith', 'david', 'degree', 'male', 10, 'married')
INSERT INTO colonist(first_name, last_name, sur_name, qualification, gender, number_of_members, civil_status)
VALUES('kamal', 'hearth', 'sumith', 'master', 'male', 5, 'unmarried')
INSERT INTO colonist(first_name, last_name, sur_name, qualification, gender, number_of_members, civil_status)
VALUES('erandha', 'hearth', 'bandra', 'phd', 'male', 6, 'unmarried')
INSERT INTO colonist(first_name, last_name, sur_name, qualification, gender, number_of_members, civil_status)
VALUES('hansika', 'banu', 'asaaa', 'hd', 'female', 3, 'unmarried')

SELECT * FROM colonist

```

The results grid shows the inserted data:

colonist_ID	first_name	last_name	sur_name	dob	qualification	age	gender	number_of_members	civil_status	
1	john	smith	david	NULL	degree	NULL	male	10	married	
2	101	hansika	banu	asaaa	NULL	NULL	female	3	unmarried	
3	111	kamal	hearth	NULL	master	NULL	male	5	unmarried	
4	123	john	smith	david	degree	NULL	male	10	married	
5	222	john	smith	david	NULL	degree	NULL	male	10	married
6	333	kamal	hearth	sumith	NULL	master	NULL	male	5	unmarried
7	444	erandha	hearth	bandra	NULL	phd	NULL	male	6	unmarried
8	456	kamal	hearth	sumith	NULL	master	NULL	male	5	unmarried
9	555	hansika	banu	asaaa	NULL	hd	NULL	female	3	unmarried
10	789	erandha	hearth	bandra	NULL	phd	NULL	male	6	unmarried

Below the results grid, a message states: "Query executed successfully."

Figure 56– create table colonist

Insert data into dependent table

The screenshot shows the Microsoft SQL Server Management Studio interface. In the Object Explorer, the 'dependent' table is selected under the 'dbo.dependent' schema. A query window titled 'SQLQuery2.sql' contains the following T-SQL code:

```

-- Create table dependent
CREATE TABLE dependent(
    dependent_ID INT IDENTITY(1,1) PRIMARY KEY,
    name VARCHAR(100),
    gender VARCHAR(100),
    relationship VARCHAR(100)
);

-- Insert data into dependent
INSERT INTO dependent(name, gender, relationship)
VALUES('john', 'male', 'blood')

INSERT INTO dependent(name, gender, relationship)
VALUES('kamal', 'male', 'blood')

INSERT INTO dependent(name, gender, relationship)
VALUES('erandha', 'male', 'child')

INSERT INTO dependent(name, gender)
VALUES('hansika', 'female')

```

The results grid shows the inserted data:

dependent_ID	name	gender	relationship
1	john	male	blood
2	kamal	male	blood
3	erandha	male	child
4	hansika	female	

Below the results grid, a message states: "Query executed successfully."

Figure 57– create table dependent

Insert data into E_jet table

The screenshot shows the Microsoft SQL Server Management Studio interface. A query window titled 'SQLQuery1.sql - DESKTOP-QDAQFDK\SQLEXPRESS.e_spacesolution (DESKTOP-QDAQFDK\NewGen (S))' is open. The code entered is:

```
use e_spacesolution
insert into Ejet(jet_ID,no_passenger_seats,nuclear_engine_power,made_year,weight,power_source)values(004,50,'100cc',2020,800,'petrol');
insert into Ejet(jet_ID,no_passenger_seats,nuclear_engine_power,made_year,power_source)values(005,80,'200cc',2020,'petrol');
insert into Ejet(jet_ID,nuclear_engine_power,made_year,weight,power_source)values(006,'300cc',2020,800,'petrol');
insert into Ejet(jet_ID,no_passenger_seats,made_year,weight,power_source)values(007,50,2020,800,'petrol');
SELECT * FROM Ejet
```

The results pane shows the inserted data:

jet_ID	no_passenger_seats	nuclear_engine_power	made_year	power_source
4	50	100cc	2020	petrol
5	80	200cc	2020	petrol
6	50	300cc	2020	petrol
7	50	NULL	2020	petrol

At the bottom, the status bar indicates 'Query executed successfully.' and '0 rows'.

Figure 58– create table e_jet

Insert data into astronomer Table

The screenshot shows the Microsoft SQL Server Management Studio interface. A query window titled 'SQLQuery1.sql - DESKTOP-QDAQFDK\SQLEXPRESS.e_spacesolution (DESKTOP-QDAQFDK\NewGen (S))' is open. The code entered is:

```
use e_spacesolution
insert into Astronomer(Astronomer_ID,space_hours,qualification)VALUES(997,48,'MASTERS');
insert into Astronomer(Astronomer_ID,space_hours,qualification)VALUES(985,49,'PHD');
insert into Astronomer(Astronomer_ID,space_hours)VALUES(588,48);
insert into Astronomer(Astronomer_ID,qualification)VALUES(987,'PHD');
```

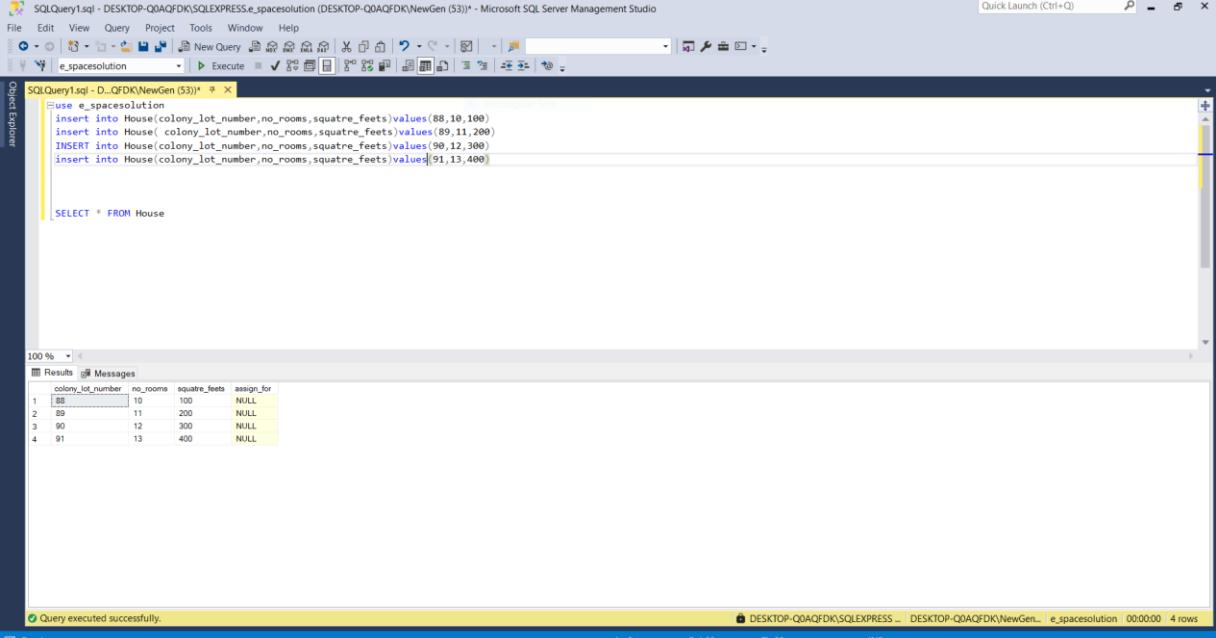
The results pane shows the inserted data:

Astronomer_ID	space_hours	qualification
997	48	MASTERS
985	49	PHD
588	48	
987		PHD

At the bottom, the status bar indicates 'Query executed successfully.' and '0 rows'.

Figure 59– create table astronomer

Insert data into house table



The screenshot shows the Microsoft SQL Server Management Studio interface. In the Object Explorer, a database named 'e_spacesolution' is selected. In the center pane, a query window titled 'SQLQuery1.sql - DESKTOP-Q0AQFDK\SQLEXPRESS.e_spacesolution (DESKTOP-Q0AQFDK\NewGen (53))' contains the following SQL code:

```

use e_spacesolution
insert into House(colony_lot_number,no_rooms,square_feet)values(88,10,100)
insert into House(colony_lot_number,no_rooms,square_feet)values(89,11,200)
INSERT into House(colony_lot_number,no_rooms,square_feet)values(90,12,300)
insert into House(colony_lot_number,no_rooms,square_feet)values(91,13,400)

SELECT * FROM House

```

Below the code, the results pane displays a table with four rows of data:

colony_lot_number	no_rooms	square_feet	assign_for
88	10	100	NULL
89	11	200	NULL
90	12	300	NULL
91	13	400	NULL

At the bottom of the interface, a status bar indicates: 'Query executed successfully.' and 'Ready'.

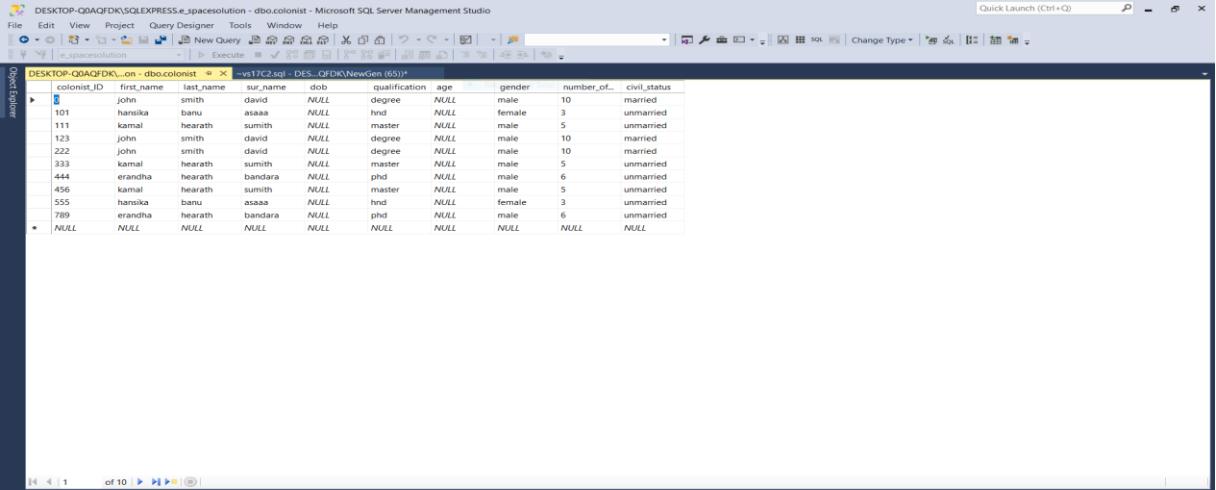
Figure 60– create table house

2nd method

The 2nd method is the easiest method to insert data is **RIGHT CLICK THE TABLE THAT YOU**

WANT TO INSERT DATA > SELECT “EDIT TOP 200 ROWS” and this will help to modify the existing data’s

Insert data into colonist table



The screenshot shows the Microsoft SQL Server Management Studio interface. In the Object Explorer, a database named 'e_spacesolution' is selected. In the center pane, a query window titled 'DESKTOP-Q0AQFDK\SQLEXPRESS.e_spacesolution - dbo.colonist - Microsoft SQL Server Management Studio' contains the following SQL code:

```

CREATE TABLE [dbo].[colonist] (
    [colonist_ID] [int] IDENTITY(1,1) NOT NULL,
    [first_name] [varchar](50) NOT NULL,
    [last_name] [varchar](50) NOT NULL,
    [sur_name] [varchar](50) NOT NULL,
    [dob] [date] NOT NULL,
    [qualification] [varchar](50) NOT NULL,
    [age] [int] NOT NULL,
    [gender] [char](1) NOT NULL,
    [number_of_...]
)

```

Below the code, the results pane displays a table with 10 rows of data:

colonist_ID	first_name	last_name	sur_name	dob	qualification	age	gender	number_of_...	civil_status
101	john	smith	david	NULL	degree	NULL	male	10	married
111	hansika	bandu	asaaa	NULL	hnd	NULL	female	3	unmarried
123	kamal	hearith	sumith	NULL	master	NULL	male	5	unmarried
232	john	smith	david	NULL	degree	NULL	male	10	married
333	kamal	hearith	sumith	NULL	master	NULL	male	5	unmarried
444	erandha	hearith	bandara	NULL	phd	NULL	male	6	unmarried
456	kamal	hearith	sumith	NULL	master	NULL	male	5	unmarried
555	hansika	bandu	asaaa	NULL	hnd	NULL	female	3	unmarried
789	erandha	hearith	bandara	NULL	phd	NULL	male	6	unmarried
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

At the bottom of the interface, a status bar indicates: 'of 10 | < > < > < > < > | Ready'.

Figure 61– create table colonist

Insert data into dependent table

The screenshot shows the Microsoft SQL Server Management Studio interface. The title bar reads "DESKTOP-Q0AQFDK\SQLEXPRESS.e_spacesolution - dbo.dependent - Microsoft SQL Server Management Studio". The main window displays a table named "dependent" with the following data:

dependent_id	name	dob	gender	relationship	colonist_ID
101	erandha	NULL	male	child	NULL
121	erandha	NULL	male	child	NULL
444	john	NULL	male	blood	NULL
445	john	NULL	male	blood	NULL
534	hansika	NULL	female	NULL	NULL
594	hansika	NULL	female	NULL	NULL
876	kamal	NULL	male	blood	NULL
888	kamal	NULL	male	blood	NULL
*	NULL	NULL	NULL	NULL	NULL

Figure 62– create table dependent

Insert data into astronomer table

The screenshot shows the Microsoft SQL Server Management Studio interface. The title bar reads "DESKTOP-Q0AQFDK\SQLEXPRESS.e_spacesolution - dbo.Astronomer - Microsoft SQL Server Management Studio". The main window displays a table named "astronomer" with the following data:

astronomer_id	space_hours	qualification	designation	rank	jet_ID
57	NULL	HND	NULL	NULL	001
58	40	NULL	NULL	NULL	NULL
97	NULL	HND	NULL	NULL	NULL
567	40	phd	NULL	NULL	NULL
588	40	NULL	NULL	NULL	NULL
657	NULL	HND	NULL	NULL	NULL
905	49	PHD	NULL	NULL	NULL
955	49	PHD	NULL	NULL	NULL
985	49	PHD	NULL	NULL	NULL
987	40	MASTERS	NULL	NULL	NULL
997	40	MASTERS	NULL	NULL	NULL
*	NULL	NULL	NULL	NULL	NULL

Figure 63– create table astronomer

Insert data into house table

The screenshot shows the Microsoft SQL Server Management Studio interface. The title bar reads "DESKTOP-Q0AQFDK\SQLEXPRESS.e_spacesolution - dbo.House - Microsoft SQL Server Management Studio". The main area displays a table named "House" with four columns: "colony_id", "no_rooms", "square_feet", and "assign_for". The data in the table is as follows:

colony_id	no_rooms	square_feet	assign_for
88	10	100	NULL
89	11	200	NULL
90	12	300	NULL
91	13	400	NULL
*	NULL	NULL	NULL

Figure 64– create table astronomer

SUITABLE IDE FOR GIVEN SCENARIO

IDE (Integrated Development Environment)

The multipurpose center that is an integrated development environment (IDE) makes it easy to understand the complex process of creating software. Essentially, an IDE combines a number of essential tools for programmers, including a powerful text editor that improves code readability with features such as autocompletion and syntax highlighting. In addition, this integrated platform includes debugging tools that make it easier to find and correct coding errors, leading to more reliable and transparent program performance. In addition, the IDE provides a seamless connection to interpreters or compilers, allowing human-readable code to be converted into executable formats and bridging the gap between functional software applications and abstract programming logic. In addition to these basic functions, many modern IDEs can be used to integrate version control systems, facilitate team collaboration in coding, and provide project management tools that make it easier to organize and coordinate different processes. Ultimately, the IDE greatly improves the development workflow by combining these essential tools and functions into a single interface. This gives developers the ability to create, modify and update software programs more accurately and efficiently.

Examples of Integrated Development Environment (IDE)

Eclipse: This open source integrated development environment (IDE) for Java is known for its adaptability. Plugins can also be used to support other languages. There are many features that you can use for personal use, such as debugging, code completion, and a wide ecosystem of plugins.

Visual Studio: Microsoft created a program called Visual Studio that can work in many programming languages, including C, C++, VB.NET, and others. There are many tools for debugging, writing code, and building Windows, web, mobile, and cloud applications.

PyCharm: It has built-in tools for testing and debugging Python applications, as well as intelligent code help and code navigation, which are intended to be used only in the context of Python development.

IDEA IntelliJ: The most widely used Java programming language, the system offers powerful refactoring tools, advanced code completion and more robust coding support. They are also supported by other languages such as Scala, Groovy and Kotlin.

XCode: XCode is designed for development on Apple's iOS and Mac computers, which means it is compatible with Objective C and Swift. It is a set of tools that allows you to create, evaluate and repair Apple application programs.

Features of IDE (Integrated Development Environment)

Compiler/interpreter integration: Developers can compile code and check for errors in the development environment using integrated development environments (IDEs), which often include compilers or interpreters for the respective programming languages.

Version control integration: Developers can manage code changes, work with teams, and track revisions directly from the IDE when version control systems like Git are supported. **Integration with third-party tools:** To access additional features such as code analysis tools, test frameworks, and deployment options, integrated development environments (IDEs) are often integrated with third-party tools and plugins.

Customization and extensibility: Many IDEs allow users to change layouts, themes, and shortcuts to suit their tastes. To provide additional functionality or support for other languages and frameworks, they also support plugins and extensions

Debugger: Debugging tools allow programmers to find and fix coding problems. It contains features such as real-time error detection, variable inspection, step-by-step code navigation, and breakpoints.

Code Editor: The core component of an IDE, the code editor improves code readability and efficiency by providing features such as syntax highlighting, code completion, and automatic indentation.

SUITABLE IDE FOR GIVEN SCENARIO

For **E_SPACE SOLUTION** we used **Visual Studio IDE**

Because Visual Studio is a highly advanced developer tool that allows you to finish the whole development cycle in a single location. It is a complete integrated development environment (IDE) that allows you to write, modify, debug, build, and publish code. Aside from code editing and debugging, and it contains graphical designers, compilers, code completion tools, source control, extensions, and a plethora of other capabilities to help with every stage of the software development process.

❖ Properties

The Properties window allows you to review and modify design attributes and events of selected objects in editors and designers. The Properties panel can also be used to edit and inspect the properties of files, projects and solutions. The Properties window is located under the View menu. You can also access it by pressing F4 or typing Properties in the search bar.

Depending on the needs of a specific property, the Properties window displays several types of edit fields. Edit boxes, drop-down lists, and links to custom editor dialogs are among the available edit fields.

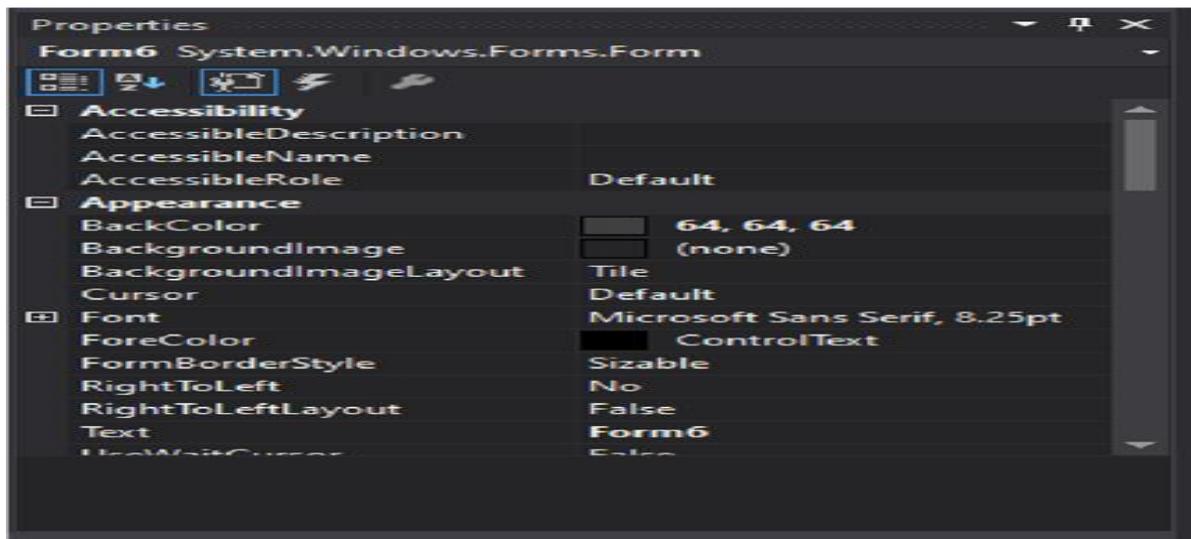


Figure 65– create table astronomer

❖ Solution Explorer

The Solution Explorer window displays a list of all the projects and files they include. The window is used to navigate to files, include/exclude files from the project, and manage project references and properties.

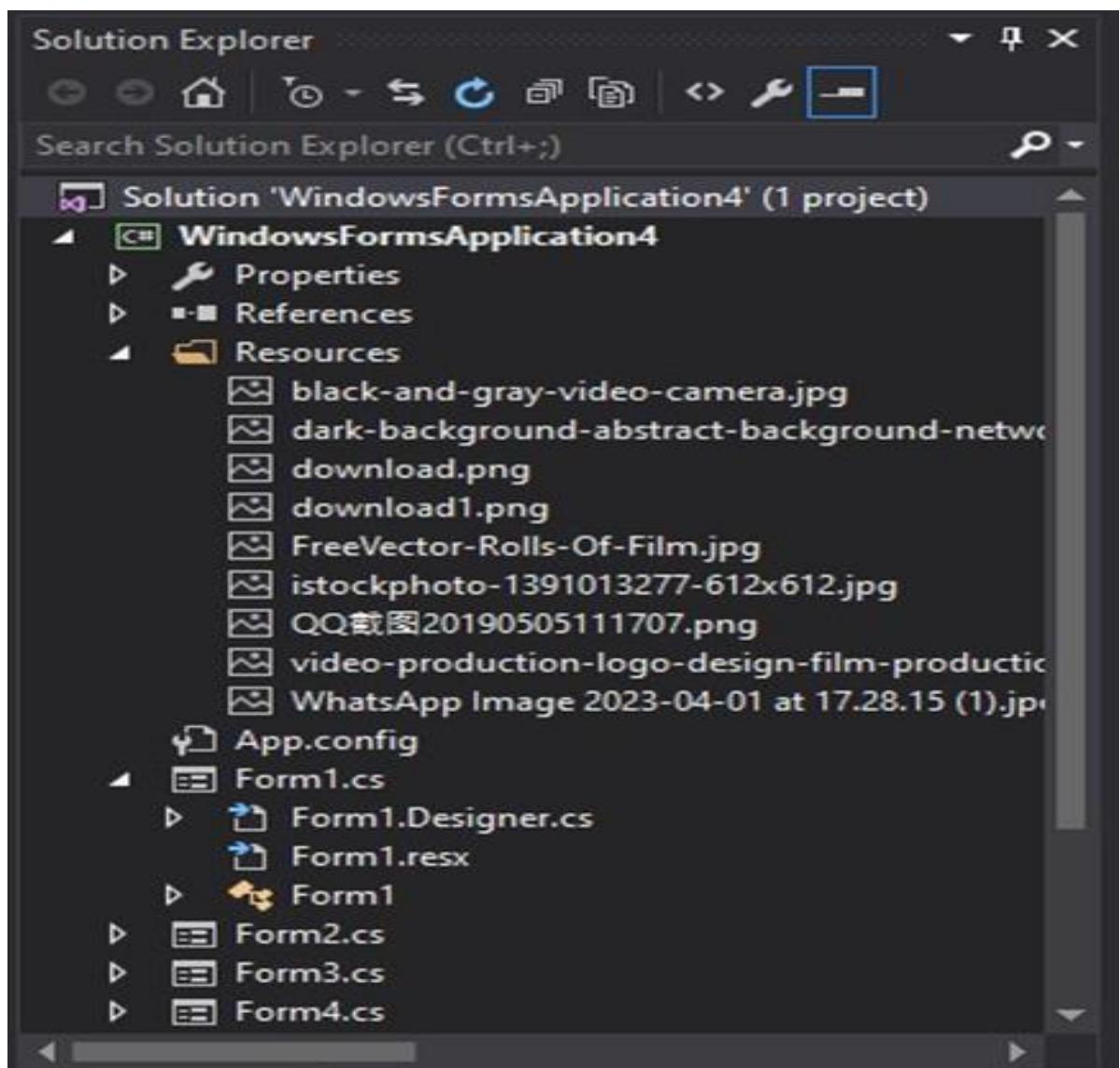


Figure 66—create table astronomer

❖ Server Explorer

Server Explorer is a shortcut that allows you to access servers that are either installed on the system or linked to it. These are often database servers like SQL Server. By visiting the server, you have access to all of the databases on that specific server, from which you can then establish the connections required within your software.

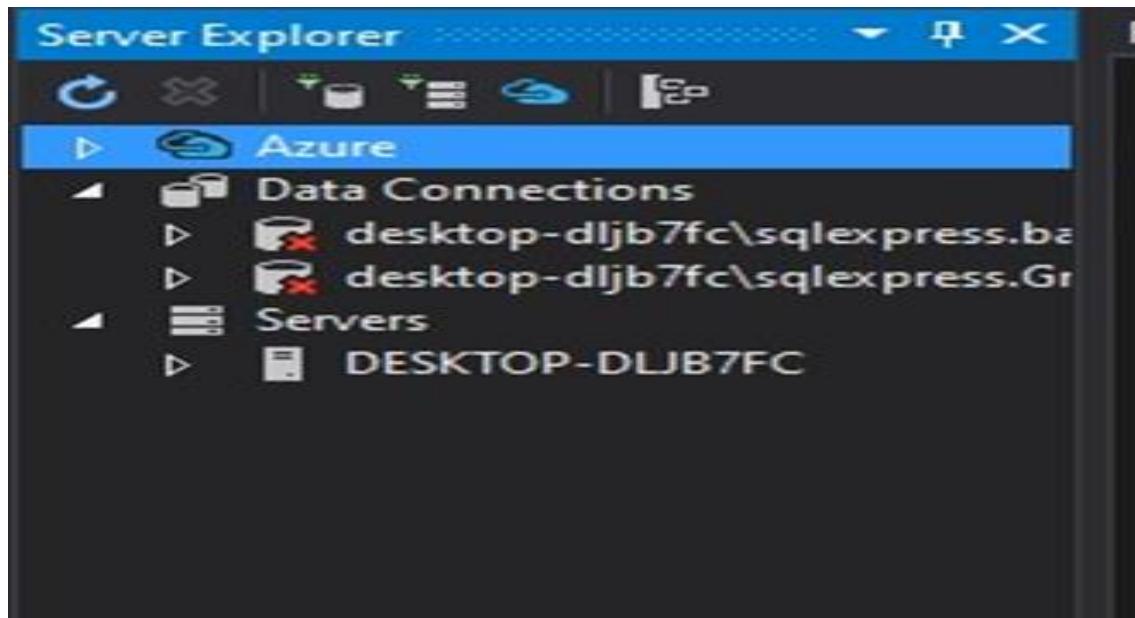


Figure 67– create table astronomer

❖ Toolbox

A Windows control is a graphical item that allows the user to interact with the computer. The controls are as diverse as the demands and aims. Because there are so many controls for varied purposes, their incorporation into an application and setting are left to the computer programmer. The Toolbox is the accessory that contains the majority of the controls used in an application:

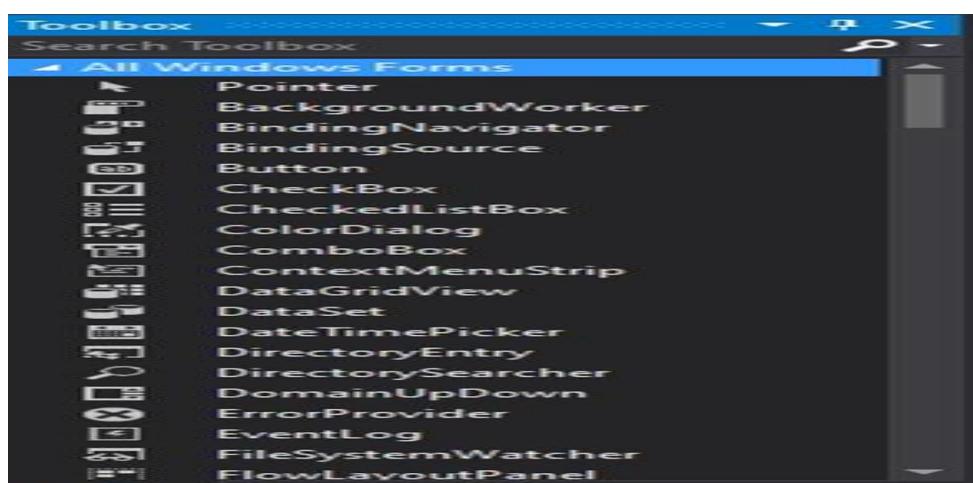


Figure 68– create table astronomer

Interfaces E_space solution private limited

Login page for E_space solution

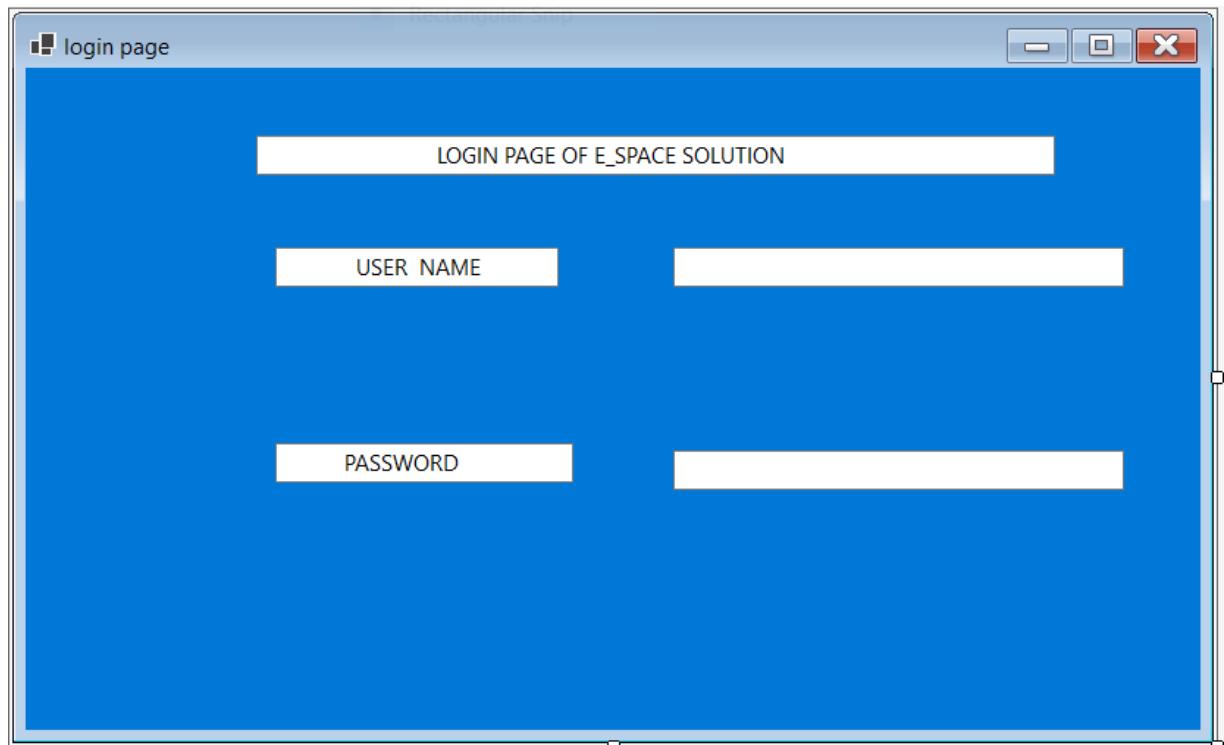


Figure 69– create table astronomer

Home page for E_space solution

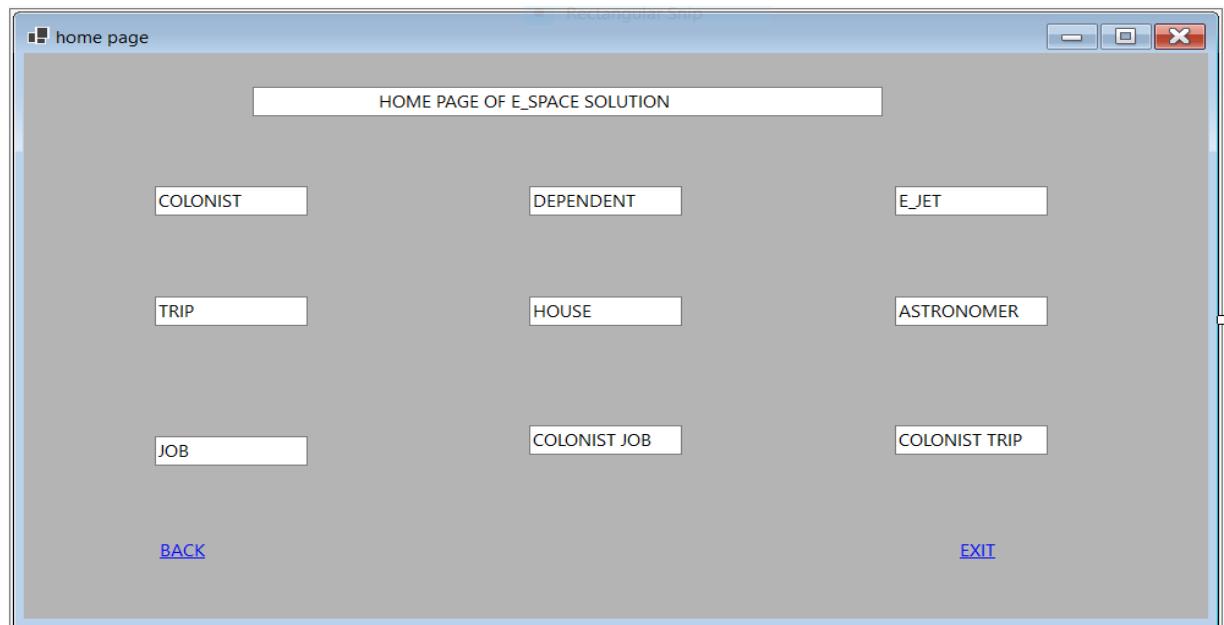


Figure 70– create table astronomer

Colonist interface for E_space solution

The screenshot shows a Windows-style application window titled "colonist page". The title bar has standard minimize, maximize, and close buttons. The main area is titled "COLONIST INTERFACE FOR E_SPACE SOLUTION". It contains five input fields in pairs, each with a label on the left and a corresponding input box on the right. The labels are: "MARS COLONIZATION ID", "NAME", "DATE OF BIRTH", "NUMBER OF FAMILY MEMBERS", and "GENDER". Below these fields are four buttons: "INSERT", "UPDATE", "DELETE", and "CLEAR". At the bottom left is a blue link labeled "EXIT", and at the bottom right is another blue link labeled "BACK".

Figure 71– create table astronomer

Dependent interface for E_space solution

The screenshot shows a Windows-style application window titled "dependent page". The title bar has standard minimize, maximize, and close buttons. The main area is titled "DEPENDENT INTERFACE FOR E_SPACE SOLUTION". It contains five input fields in pairs, each with a label on the left and a corresponding input box on the right. The labels are: "DEPENDENT ID", "COLONIST ID", "NAME", "DATE OF BIRTH", and "GENDER". Below these fields are four buttons: "CLEAR", "INSERT", "UPDATE", and "DELETE". At the bottom left is a blue link labeled "BACK", and at the bottom right is another blue link labeled "EXIT".

Figure 72– create table astronomer

E_jet interface for E_space solution

The screenshot shows a Windows application window titled "e_jet page". The title bar also includes standard window controls for minimize, maximize, and close. The main area is titled "E_JET INTERFACE OF E_SPACE SOLUTION". It contains five input fields arranged in two columns. The first column has fields for "JET ID", "MADE YEAR", "WEIGHT", "POWER SOURCE", and "NUMBER OF SEATS". The second column has corresponding empty input fields. Below the input fields are four buttons: "CLEAR", "INSERT", "UPDATE", and "DELETE". At the bottom left is a "BACK" link, and at the bottom right is an "EXIT" link.

Figure 73– create table astronomer

Trip interface for E_space solution

The screenshot shows a Windows application window titled "trip page". The title bar includes standard window controls. The main area is titled "TRIP INTERFACE OF E_SPACE SOLUTION". It contains four input fields arranged in two columns. The first column has fields for "TRIP ID", "JET CODE", "LAUNCH DATE", and "RETURN DATE". The second column has corresponding empty input fields. Below the input fields are four buttons: "CLEAR", "INSERT", "UPDATE", and "DELETE". At the bottom left is a "BACK" link, and at the bottom right is an "EXIT" link.

Figure 74– create table astronomer

House interface for E_space solution

The screenshot shows a Windows application window titled "house page". The title bar also includes standard window controls (minimize, maximize, close). The main area is titled "HOUSE INTERFACE OF E_SPACE SOLUTION". It contains three input fields arranged in a grid: "COLONY LOT NUMBER" (top-left), "NUMBER OF ROOMES" (top-right), and "SQUARE FEETS" (bottom-left). Below these fields are four buttons: "CLEAR", "INSERT", "UPDATE", and "DELETE" (top row) and "BACK" (left) and "EXIT" (right) (bottom row).

Figure 75– create table astronomer

Job interface for E_space solution

The screenshot shows a Windows application window titled "job page". The title bar includes standard window controls. The main area is titled "JOB INTERFACE OF E_SPACE SOLUTION". It contains two input fields arranged in a grid: "JOB ID" (top-left) and "JOB TITTLE" (top-right). Below these fields are four buttons: "CLEAR", "INSERT", "UPDATE", and "DELETE" (top row) and "BACK" (left) and "EXIT" (right) (bottom row).

Figure 76– create table astronomer

Security mechanisms

Database security refers to the process of securing and safeguarding a database against unauthorized access or cyber-attacks. Database security methods are want to implemented in the firm. In this post, we will cover each sort of database security strategy used to safeguard the database.

Controlling of unauthorized access

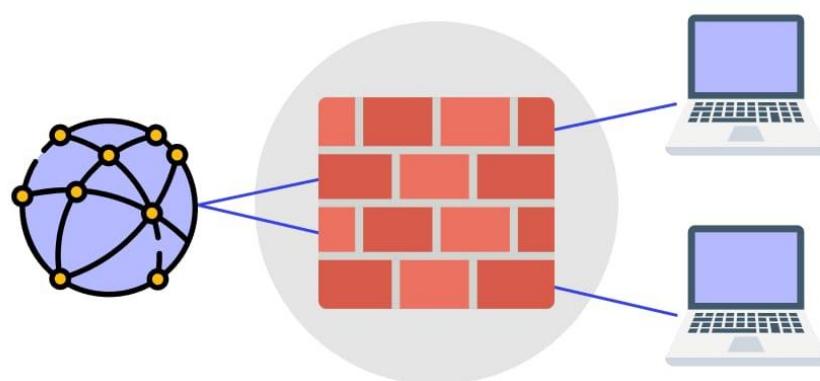
- ✓ Fire wall
- ✓ Otp (one time password)
- ✓ Strong Password
- ✓ Physical Security
- ✓ Encryption
- ✓ Digital signature
- ✓ Authentication
- ✓ Backup Database
- ✓ Application Security

✓ Firewall

Is either a h/w or s/wwhich can scan the incoming and outgoing traffic through the public h/w There are rules set by the admin to control unauthorized accessing while the incoming and outgoing traffic are scanned against the rules to detect hackingeventes.

The firewall shields your network by acting as a 24/7filtert.

The concept of a firewall



(Beschokov, 2024) Figure 77– create table astronomer

Two computers connected to the internet by using the firewall

✓ Otp (one time password)

OTP- One Time password is a code which can be used only once to access a resource (account) a different pass word has to be used for the next time to log on to the system.

There are Pros in OTP than the traditional password.

- No need to memorize keywords
- No need to consider strong pass words
- Less hacking events.
- Diffcut to guess

✓ Strong Password

This is a basic yet crucial step for ensuring database security. You should never reveal your password as a DBA or IT officer. If you choose a simple password such as your mobile number, employee ID, or date of birth, hackers will try to access using these passwords. As a result, you will lose control of your database.

Create a strong password for database login that includes a combination of letters, numbers, and special characters (minimum of 10 characters overall) and update it on a regular basis.
for example, ESEsn#\$@12# can be a strong password for your database login.

✓ Physical Security

Physical database security is the protection of the database server room against unauthorised access. A database server should be placed in a secure and climate-controlled environment.

✓ Encryption

☒ Man in the middle attack

When sensitive details (password, credit card details) are passing through public networks (internet), there are unauthorized parties (hackers) who are willing to read the above sensitive details to continuous unauthorized accessing which can be controlled by the concept: **encryption**

Encryption

Is the process of scrambling the original message (text) into a meaningless format at the sender using a key (secret) then the encrypted text (cipher text) is added to the public network to be delivered to the receiver

It may be read by (accessed by) the hackers from the public network while it is also collected by the specific receiver

It is not meaningful for the hackers but the receiver decrypts it using the key into the meaningful original text. The sender and the receiver should exchange the key (secret) before the communication continuous which should not be known to the other parties

The basics encryption method is chipper text

It is shifting the alphabet by same letters so that the no.of.shifts become the key .while replacing the database in to original text by the new alphabet created. The decryption of the receiver continuous in the reverse order using the same key and creating the original plain text to deliver the msg.

Furthermore methods for encryption are two types:

- Symmetric key encryption
- Asymmetric key encryption

Symmetric key encryption

Symmetric key encryption uses the same key is used at the sender and the receiver to encrypt and decrypt the electronic database . If you encrypt a zip file and then decrypt with the same key, you are using symmetric encryption. Symmetric encryption is also called “secret key” encryption and the key must be kept secret from third parties. The entities communicating via symmetric encryption must exchange the key so that it can be used in the decryption process and their for the exchange method medium such as (SMS, telephone call, physical meetings) can be used by the sender and the receiver to share the same key.

Symmetric Encryption

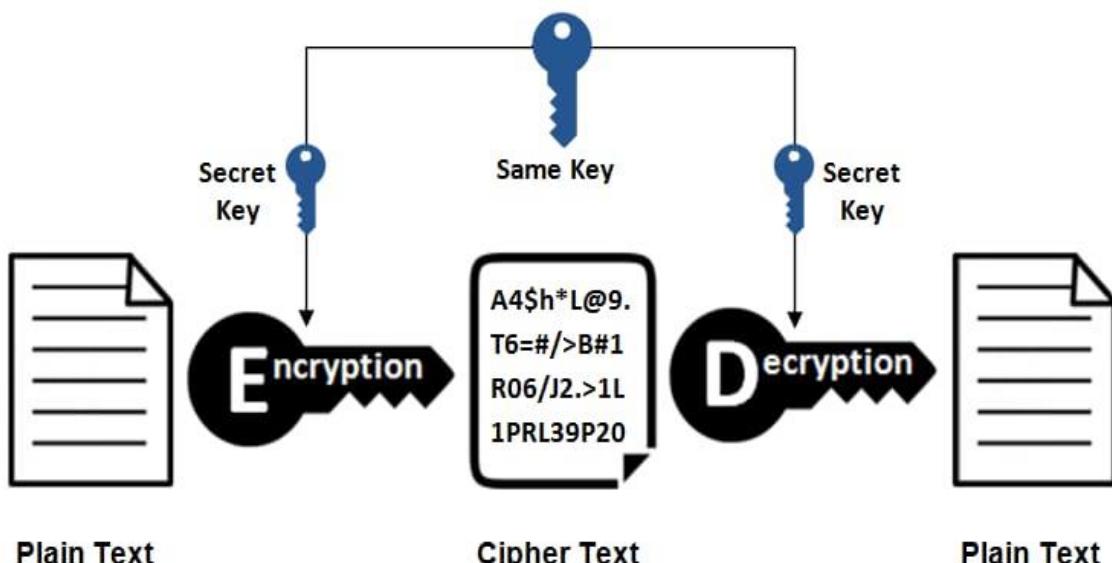


Figure 78

A symmetric key encryption

A solution for the problem of the symmetric key encryption. A symmetric key encryption can be classified as :

- **Public key encryption**
- **Private Key encryption**

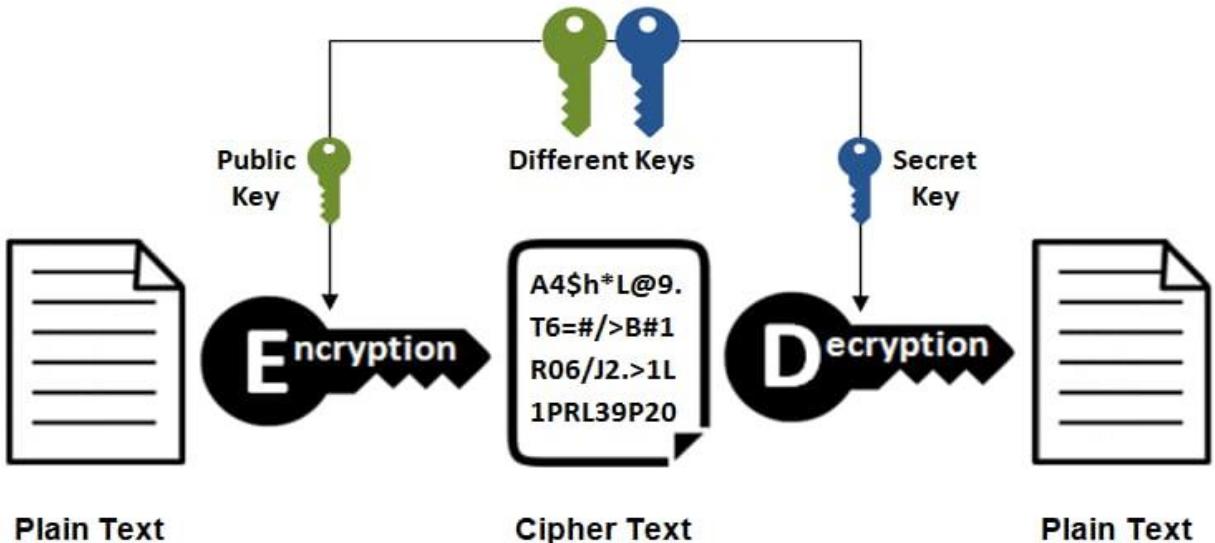
Public key encryption

Public key are known to all which can be announced on website. This is the key used for encrypting or signing the message

Private Key encryption

Secret which must only be known by the owner of the private key. At first the receiver should register for the public key and private key. When registering for public key encryption the public and private key issued which are having a connection among them the system knows the matching set of public and private key. There are some public key encryption software to support the encryption and decryption activity. Whenever text is encrypted using a public key it can only be decrypted using the matching private key of the public. No other private key can decrypt the message. Receiver must register for the public key encryption and obtain public and private key.

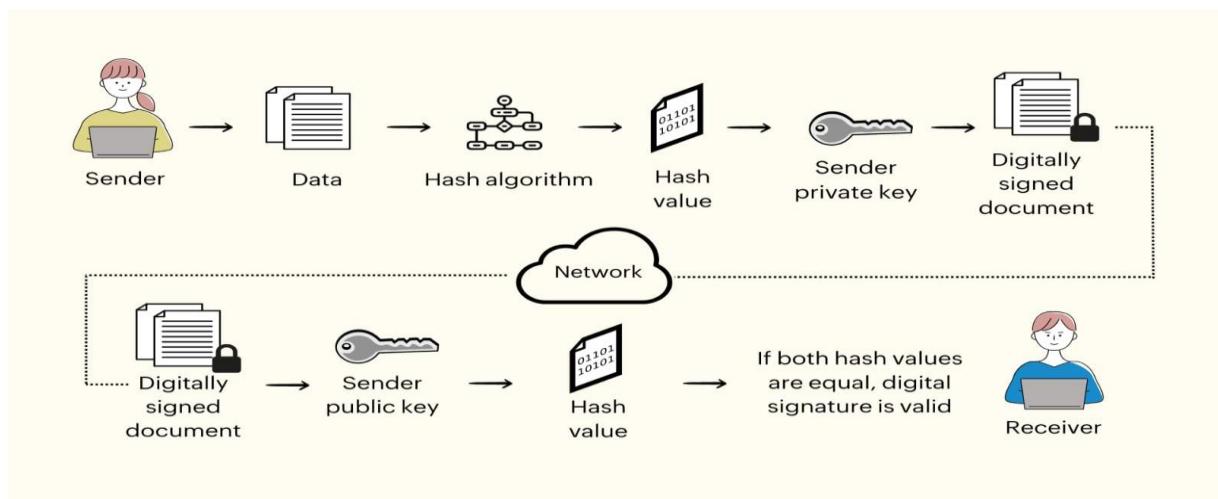
Asymmetric Encryption



(Gupta, 2024) Figure 79

✓ Digital signature

Digital signature is a replacement for the traditional signature in the digital medium which is legally accepted to provide the authentication. When using digital signatures it is a must to provide the specific public key of the person to the receiving party to carry out the verification. The agreed text is encrypted using the private of the sender and create the digital signatures. It is added to the document and exchange it with the receiver the receiver maintain the list of public key of the sender.



(padmalosani, 2023) *Figure 80—create table astronomer*

✓ Authentication

Database authentication is a type of database security that verifies user login information stored in the database. If the user's login information matches that of the database, the user can access the database. This means that the user is authorized to access your database.

If a real user has a certain level of data access privilege, he cannot access any other data. No unauthorized or malicious person can access your database. Therefore, database authentication is essential to ensure database security.

✓ Backup Database

Backup is another sort of database protection that is used to recover data in the event of data loss, data corruption, hacking, or natural catastrophes. It copies or archives the database in real time on secondary storage.

If you configured the primary and secondary servers in the same location, and the primary server is destroyed, the secondary server may also be destroyed. So you can't execute your program, and your machine will shut down until you recover.

As a result, it is recommended that the backup server be physically configured in a distinct location in order to maintain database security. If the primary server fails, you can retrieve the database from the secondary server.

✓ Application Security

To defend yourself from online assaults like SQL injection, you must secure your application and database. SQL injection is the most prevalent online attack in which hackers manage an application's database in order to hack sensitive information or damage the database.

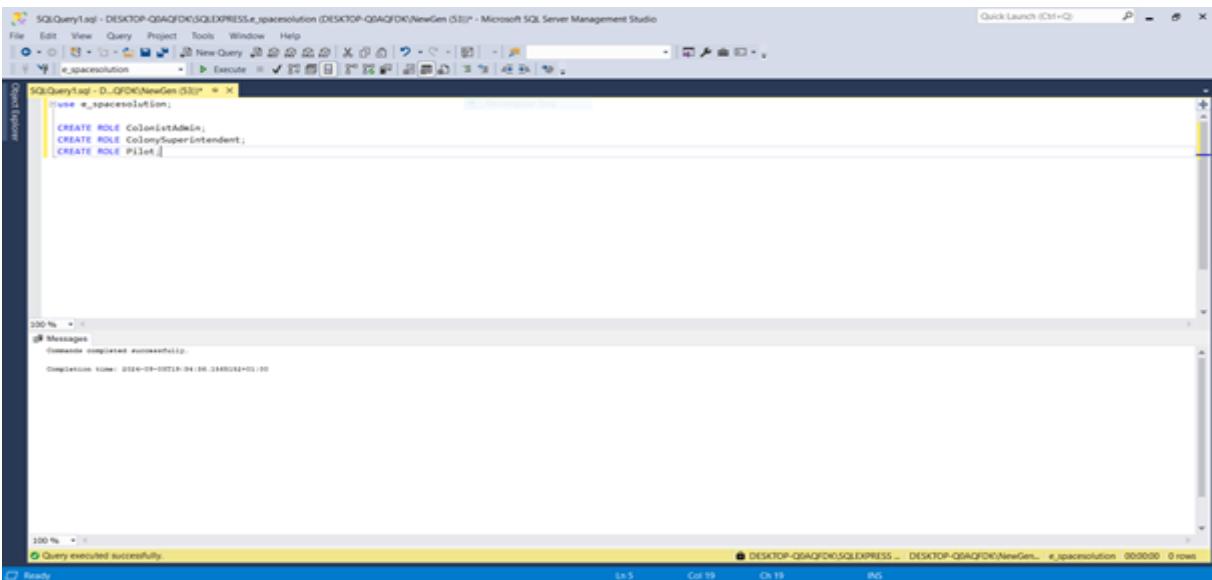
In this approach, the attacker inserts malicious code into a SQL query using web page input. It occurs when an application fails to appropriately sanitize SQL statements. As a result, the attacker can add their own malicious SQL statements to access your database for malevolent purposes.

You may safeguard your application against SQL injection attacks by using the following preventative measures:

- o Using Prepared Statements
- o Use a Web Application Firewall.
- o Updating your system
- o Validating user input
- o Limiting privileges
- o Use Stored Procedures

Implementing a security mechanisms in sql server

Create user groups



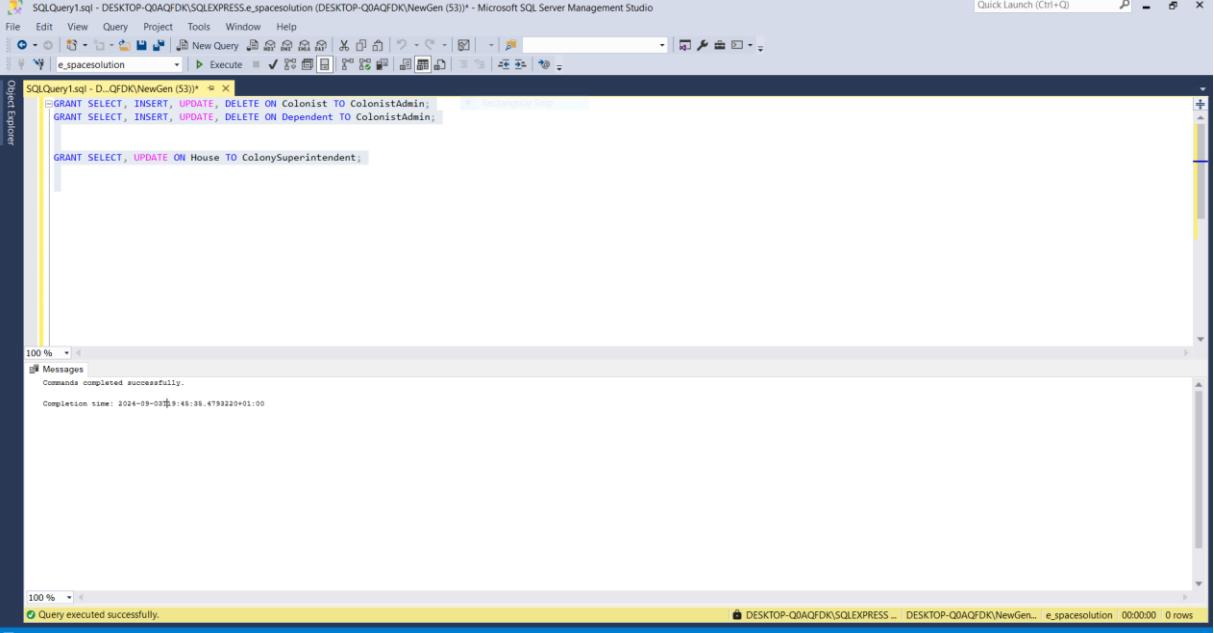
The screenshot shows the Microsoft SQL Server Management Studio interface. A query window titled 'SQLQuery1.sql' is open, displaying the following T-SQL code:

```
use e_spacesolution;
CREATE ROLE ColonyAdmin;
CREATE ROLE ColonySuperintendent;
CREATE ROLE Pilot;
```

The 'Messages' pane at the bottom left shows the command completed successfully with a timestamp of 2024-09-03T19:04:06.1840000+01:00. The status bar at the bottom right indicates '0 rows'.

Figure 81– create table astronomer

Assign permissions



```
GRANT SELECT, INSERT, UPDATE, DELETE ON Colonyist TO ColonistAdmin;
GRANT SELECT, INSERT, UPDATE, DELETE ON Dependent TO ColonistAdmin;

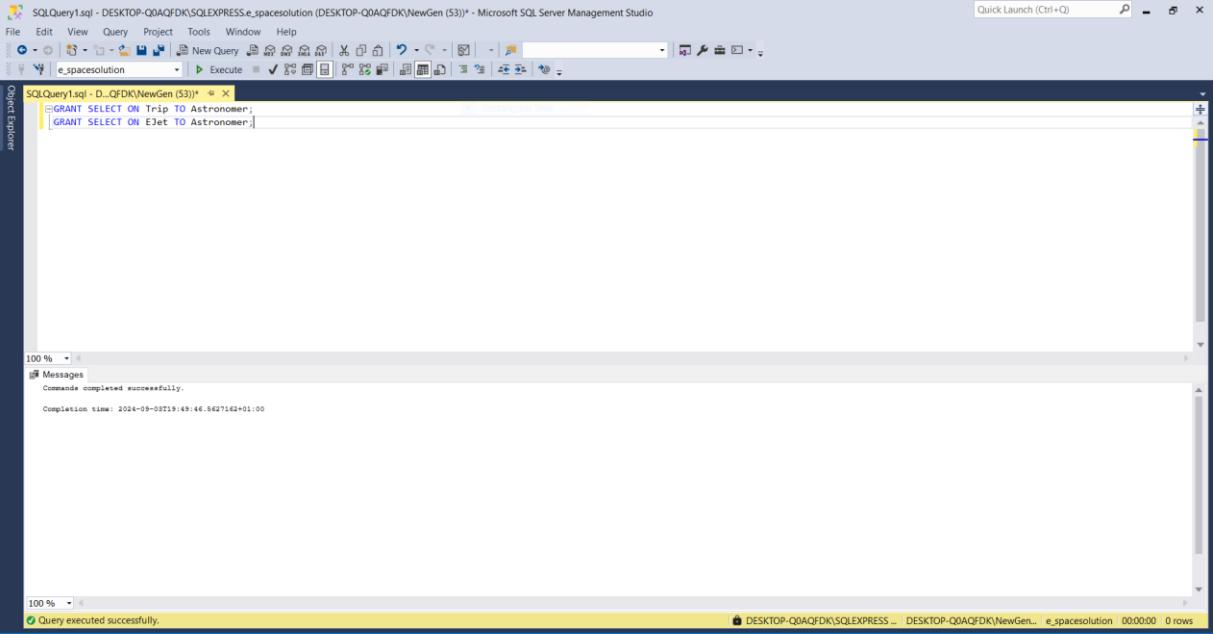
GRANT SELECT, UPDATE ON House TO ColonySuperintendent;
```

Messages
Commands completed successfully.
Completion time: 2024-09-03T09:46:38.4793220+01:00

Query executed successfully.

Figure 82– create table astronomer

Assign user to roles



```
GRANT SELECT ON Trip TO Astronomer;
GRANT SELECT ON EJet TO Astronomer;
```

Messages
Commands completed successfully.
Completion time: 2024-09-03T19:46:56Z7162+01:00

Query executed successfully.

Figure 83– create table astronomer

3. Activity

Testing the database system

Test the system process

A test plan is a written document outlining the objectives, scope, strategy, resources, schedule, and criteria for testing a certain project or product. Serves as a testing guide, ensuring that testing procedures are well-organised and scheduled.

Objectives of test plan for E_SPACE SOLUTION

By usage of test plan we can improve the database accurately when storing all necessary data of colonists, dependent, e-jets, trips, houses and jobs etc.... . The test plan validate the data integrity, consistency, accuracy are maintain through the test plan in the database. The most needed function of the test plan is supports the future reporting needs as well verify all the functional requirement. Test plan of the database against the both user and system requirement.

Functions of the system

- ⊕ The user must be able to access the system by logging in with their username, and password.
- ⊕ From the main menu interface, we can easily select categories.
- ⊕ It should be able to update database data.
- ⊕ The user can enter data into the system as a input.
- ⊕ It should be capable of erasing data from the system.
- ⊕ It should be able to search the database for information.
- ⊕ The GUI-based interfaces should be simple to use.

The purpose of test plan

- ❖ Login to the system.
- ❖ Insert data from the system.
- ❖ Delete data from the system.
- ❖ Searching information from the system.
- ❖ Update data from the system.

Types of testing

There are 4 types of testing there are,

Unit testing

A unit test starts with testing the individuals program of a software products.software developers and QA staff write the unit testing during the development process.

Integration testing

Integration testing, also known as chain testing or wire testing, is the process of combining various application modules and testing their behaviour as a combined or integrated unit. It is critical to ensure that the individual units communicate appropriately with one another and work properly after integration.

System testing

System testing is a type of software testing that evaluates the operation and performance of a complete and integrated software system. They determine if the system meets the given standards and is suitable for delivery to the end user. This type of testing takes place after the integration test, but before the acceptance test.

Acceptance testing

Acceptance testing is an important part of software testing that helps ensure that the software meets user needs and business objectives. The main objective of this test is to assess the compatibility of the system with the needs of the companies and determine if it is suitable for delivery.

Levels of testing for this system

Because this system deals with Graphical User Interfaces (GUIs), the author ran validation tests. In this scenario, we will test the login page as well as the capability for inserting, updating, and deleting. Validation testing is the best testing approach in this scenario because these features are incorporated into the interface. This level of testing allows us to readily identify flaws and design the best possible system based on user requirements.

Method of testing for this system

The author tested this system using black box testing methodology. Black box testing is a technique used to evaluate system functionality without access to internal connections, complex details, or the underlying source code. Black box testing primarily examines the input and output of the system. The graphical user interface (GUI) in our scenario has been tested only by the author. Black box testing is the most effective testing strategy. External testers are also skilled at identifying bugs. This is why the author used black box testing methodology.

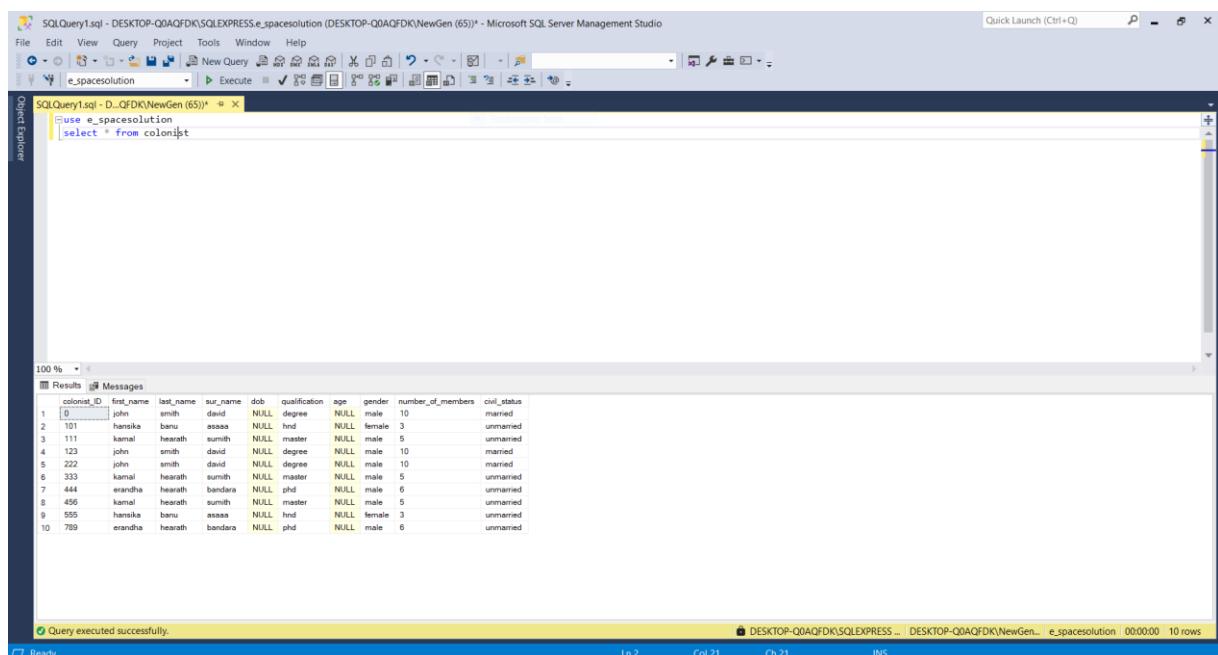
Test plan

A Test Plan is a thorough document that specifies the test strategy, objectives, timetable, estimation, deliverables, and resources needed to accomplish software testing. The Test Plan assists us in determining the amount of work required to confirm the quality of the application under test.

Testcase number	Test case	description	Expected outcome	Actual outcome
01	Search	Search all facility details from colonist	Will show all the colonist detailes	Colonist details has been searched
02	Insert	Inserting dependent id, name , gender into dependent table	Data will be inserted	Row affected
03	Delete	Delete astronomer delete from astronomer where astronomer id =985	Astronomer will be deleted	Astronomer deleted
04	Update	Update House table colony lot number,no_room,square_feets	House table will be updated	House table updated

Testcase screenshots

01. Test Case 01 (SEARCH)



The screenshot shows the Microsoft SQL Server Management Studio interface. A query window titled 'SQLQuery1.sql' is open, displaying the following SQL code:

```
use e_spacesolution
select * from colonist
```

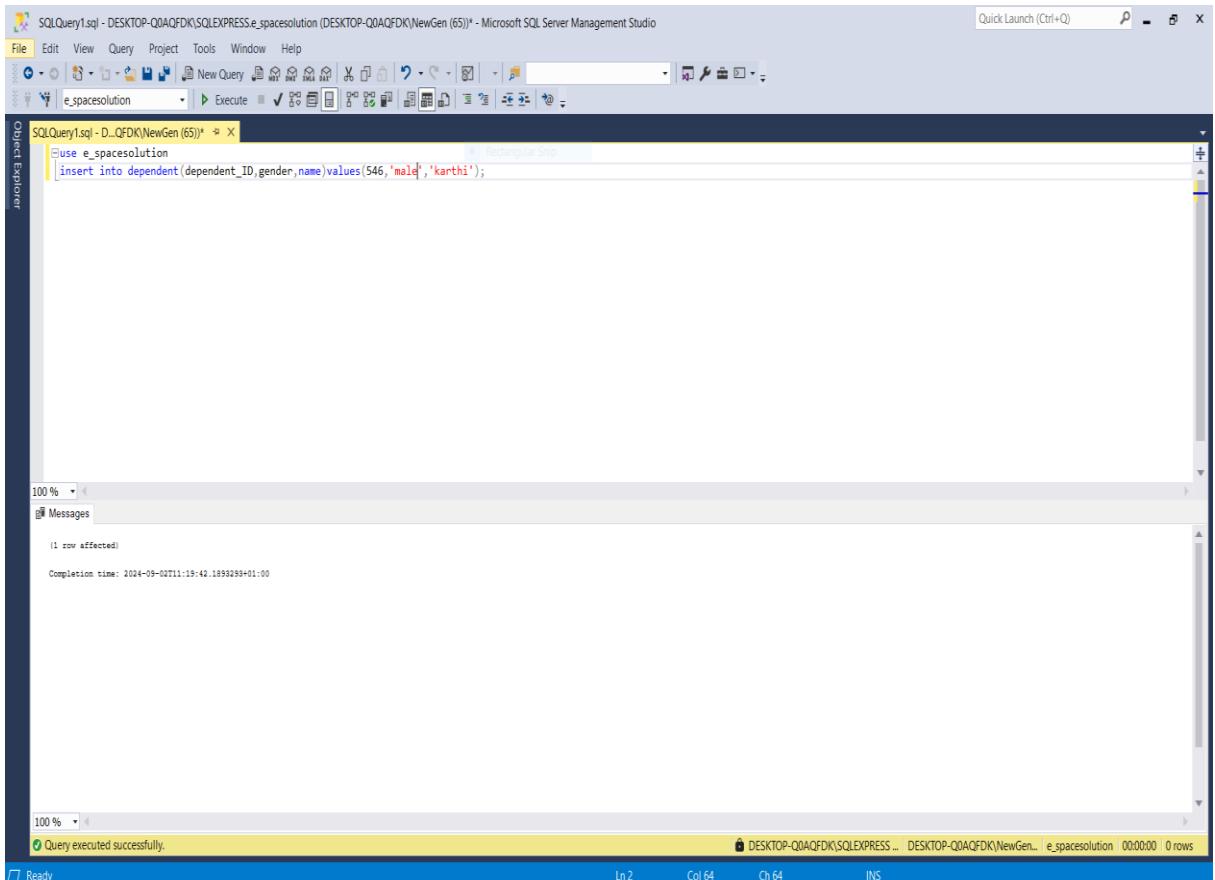
The results pane shows a table with 10 rows of data from the 'colonist' table. The columns are: colonist_ID, first_name, last_name, sur_name, dob, qualification, age, gender, number_of_members, and civil_status. The data is as follows:

colonist_ID	first_name	last_name	sur_name	dob	qualification	age	gender	number_of_members	civil_status	
1	john	smith	david	NULL	degree	NULL	male	10	married	
2	henrike	benu	assaaa	NULL	hnd	NULL	female	3	unmarried	
3	111	kamal	hearth	sumith	NULL	master	NULL	male	5	unmarried
4	123	john	smith	david	NULL	degree	NULL	male	10	married
5	222	john	smith	david	NULL	degree	NULL	male	10	married
6	333	kamal	hearth	sumith	NULL	master	NULL	male	5	unmarried
7	444	erandha	hearth	bandara	NULL	phd	NULL	male	6	unmarried
8	456	kamal	hearth	sumith	NULL	master	NULL	male	5	unmarried
9	555	henrike	benu	assaaa	NULL	hnd	NULL	female	3	unmarried
10	789	erandha	hearth	bandara	NULL	phd	NULL	male	6	unmarried

At the bottom of the results pane, it says 'Query executed successfully.'

Figure 84—create table astronomer

02. Test Case 02 (INSERT)



The screenshot shows the Microsoft SQL Server Management Studio interface. A query window titled 'SQLQuery1.sql' is open, displaying the following T-SQL code:

```
use e_spacesolution
insert into dependent(dependent_ID,gender,name)values(546,'male','karthi');
```

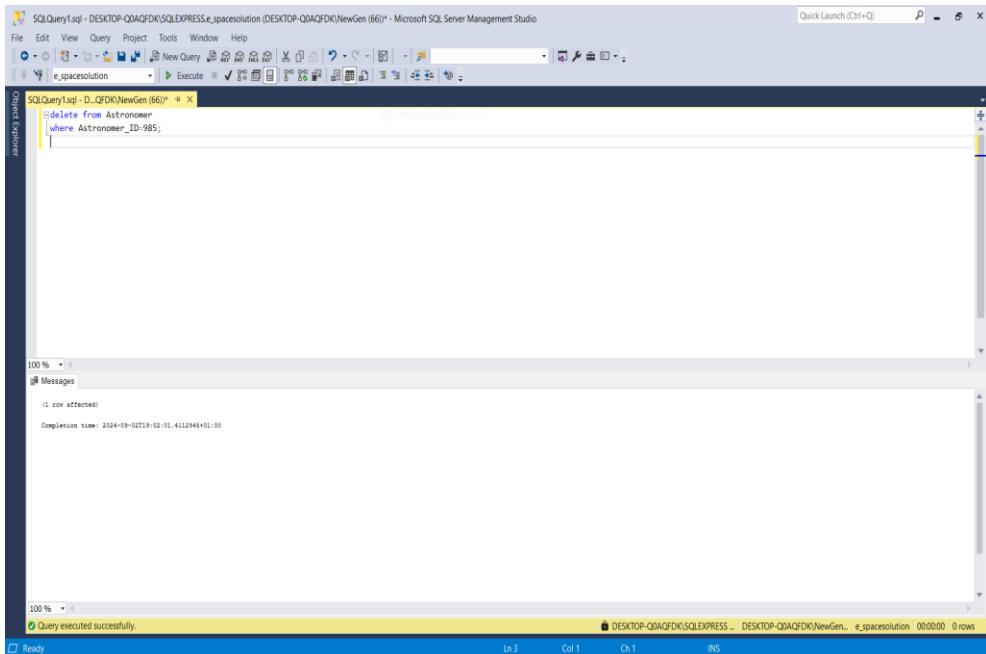
The 'Messages' pane at the bottom shows the execution results:

```
(1 row affected)
Completion time: 2024-09-02T11:19:42.1893259+01:00
```

A status bar at the bottom right indicates: DESKTOP-QQAQFDK\SQLEXPRESS .. DESKTOP-QQAQFDK\NewGen.. e_spacesolution 00:00:00 0 rows.

Figure 85—create table astronomer

03. Test Case 03 (DELETE)



The screenshot shows the Microsoft SQL Server Management Studio interface. A query window titled 'SQLQuery1.sql' is open, displaying the following T-SQL code:

```
use e_spacesolution
Delete from Astronomer
where Astronomer_ID=385;
```

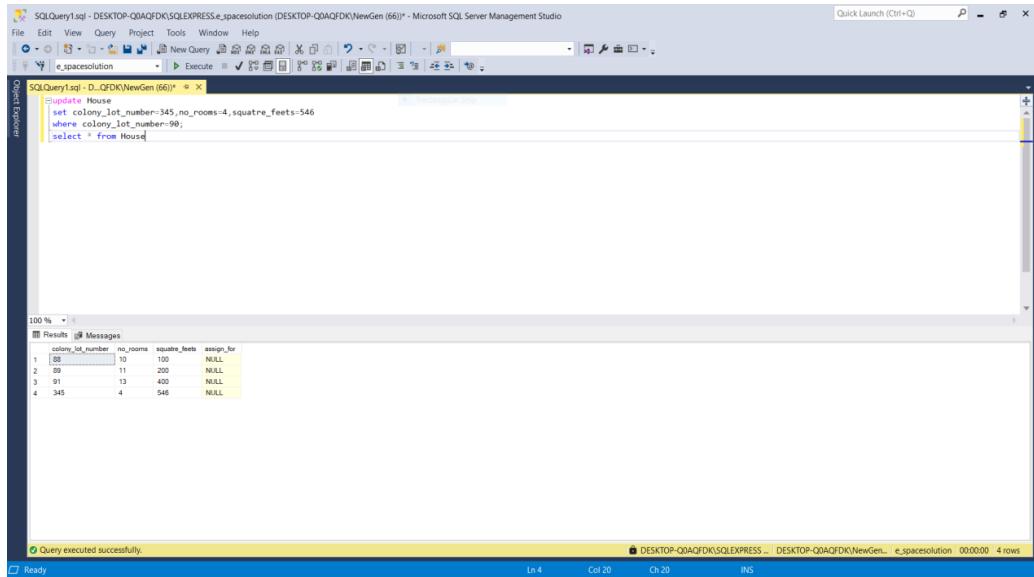
The 'Messages' pane at the bottom shows the execution results:

```
(1 row affected)
Completion time: 2024-09-02T19:02:01.412948+01:00
```

A status bar at the bottom right indicates: DESKTOP-QQAQFDK\SQLEXPRESS .. DESKTOP-QQAQFDK\NewGen.. e_spacesolution 00:00:00 0 rows.

Figure 86—create table astronomer

04. Test Case 04 (UPDATE)



The screenshot shows the Microsoft SQL Server Management Studio interface. A query window titled 'SQLQuery1.sql - DESKTOP-QOAQFDK\SQLEXPRESS.e_spacesolution (DESKTOP-QOAQFDK\NewGen (66))' contains the following SQL code:

```
Update House
set colony_lot_number=345,no_rooms=4,square_feets=546
where colony_lot_number=90;
select * from House;
```

The results pane displays the following data:

	colony_lot_number	no_rooms	square_feets	assign_for
1	345	4	546	NULL
2	90	10	100	NULL
3	91	13	200	NULL
4	91	400	NULL	NULL

A status bar at the bottom indicates: 'Ready' 'Ltr 4' 'Col 20' 'Ch 20' 'INS'. A message bar at the bottom says 'Query executed successfully.'

Figure 87– create table astronomer

TEST CASE FORM

Test Case Parameters

1. Test Case ID
2. Test function
3. Steps
4. Pre - Defined function
5. Post - Defined function
6. Date of testing
7. Expected outcome
8. Actual outcome

Test Case 01

Test Id	000
Test Function	login
Steps	<ul style="list-style-type: none"> • Enter username • Enter password
Pre-defined function	-
Post-define function	login
Date of testing	24/08/2024
Expected outcome	User should login and message box should display "Logged successfully!"
Actual outcome	 <p><i>logged in</i></p>
Tester	Mohamed Aashik

Test Case 02

Test Id	001
Test Function	login
Steps	<ul style="list-style-type: none"> • Enter username • Enter wrong password
Pre-defined function	-
Post-define function	login
Date of testing	24/08/2024
Expected outcome	User should not login and message box should display "Please enter correct username and password"
Actual outcome	 <p><i>Please enter correct username and password</i></p>
Tester	Mohamed Aashik

Test Case 03

Test Id	002
Test Function	Remove
Steps	<ul style="list-style-type: none"> Enter colonist-ID Click Search button Click delete button
Pre-defined function	Search
Post-define function	Update
Date of testing	24/08/2024
Expected outcome	User should able to delete data from the system & message box should display “data deleted”
Actual outcome	 <p><i>data deleted</i></p>
Tester	Mohamed Aashik

Test Case 04

Test Id	003
Test Function	Search
Steps	<ul style="list-style-type: none"> Enter invalid colonist- ID Click search button
Pre-defined function	colonist
Post-define function	-
Date of testing	24/08/2024
Expected outcome	User should not able to search staff details and message box should display “No results found”
Actual outcome	 <p><i>No results found</i></p>
Tester	Mohamed Aashik

Test Case 05

Test Id	004
Test Function	Insert
Steps	<ul style="list-style-type: none"> Enter all details Click add button
Pre-defined function	Login
Post-define function	Update
Date of testing	24/08/2024
Expected outcome	User should able to insert details to system and message box should display "Data inserted"
Actual outcome	 <p><i>Data inserted</i></p>
Tester	Mohamed Aashik

Testing effectiveness

More testing is required for a successful test strategy. Without these, there is nothing. The test case will not be very effective to test. I had many exam plans on my PC. Because of the test technique, my system has been greatly enhanced. My method contains numerous fundamental flaws. Test cases were used to overcome these concerns. The first test event was successfully logged without any problems, and everything was good. But In the fifth test case, it added data, and if the data was correctly entered, the interface may display the message "Data inserted." However, the computer system made a mistake. The deleting procedure likewise failed in the third test situation. I got the machine under control and fixed the problems. Before submitting or delivering the system to the client we must test it.

We can obtain further projects in the future if we have a system that is error-free. Effectiveness of testing with meaningful data

- Ensured good test coverage.
- Reusable test cases
- High-quality goods lead to more delighted consumers
- Confirmation that the program meets end-user needs.
- Improved software quality and user experience

Feedback form

Every feedback form are created by using the Google form application in Google. Google forms is a Google web based programs that allow user to design, distribute, and evaluate surveys forms. This Google form are used for the collections of data, surveys, quizzes more...

The screenshot shows a Google Form titled "E_SPACE SOLUTION". At the top, it displays the email "aashikmahroof123@gmail.com" and the status "Not shared". A note at the bottom indicates that red asterisks (*) denote required questions. The form contains three text input fields labeled "Your name *", "Email address *", and "Date *". Each field has a placeholder "Your answer" and a dropdown arrow for date selection.

Figure 88– create table astronomer

The screenshot shows a Google Form titled "create table astronomer". It features two rating scales. The first section, "How user friendly is our software", has five numbered options from 1 to 5, each preceded by an empty circle for selection. The second section, "Performance of our softsoftware", also has five numbered options from 1 to 5, each preceded by an empty circle for selection.

Figure 89– create table astronomer

Overall rating

1
2
3
4
5

Suggestions for improvement

Your answer

Feedback

Your answer

Submit **Clear form**

Never submit passwords through Google Forms.
This content is neither created nor endorsed by Google. [Report Abuse](#) - [Terms of Service](#) - [Privacy Policy](#)

Google Forms

Figure 90– create table astronomer

Non-technical people and some technologists provided feedback on the database solution.

Response 1

E_SPACE SOLUTION

aashikmahroof123@gmail.com
[Switch accounts](#)

✉ Not shared
🕒 Draft saved

* Indicates required question

Your name *
Mohammed

Email address *
mohammednafruz@gmail.com

Date *
Date
25/08/2024

Figure 91– create table astronomer

How user friendly is our software

1

2

3

4

5

[Clear selection](#)

Performance of our software

1

2

3

4

5

[Clear selection](#)

Figure 92—create table astronomer

Overall rating

1

2

3

4

5

[Clear selection](#)

Suggestions for improvement

Need more security

Feedback

Excellent |

[Submit](#) [Clear form](#)

Create survey at www.surveymonkey.com/s/ask-astronomer through Google Forms

Figure 93

Response 2

E_SPACE SOLUTION

aashikmahroof123@gmail.com
Switch accounts

Not shared
 Draft saved

* Indicates required question

Your name *

Eranga

Email address *

erangakandy123@gmail.com

Date *

Date
25/08/2024 ▾

Figure 94– create table astronomer

How user friendly is our software

1
2
3
4
5

Clear selection

Performance of our softsoftware

1
2
3
4
5

Clear selection

Figure 95

Overall rating

1

2

3

4

5

Suggestions for improvement

Everything seem to be fine

Feeback

Excellent |

Figure 96

Response 3

E_SPACE SOLUTION

aashikmahroof123@gmail.com
Switch accounts

✉ Not shared
☁ Draft saved

* Indicates required question

Your name *

Danajiya

Email address *

danajiya@gmail.com

Date *

Date
31/08/2024

Figure 97– create table astronomer

How user friendly is our software	
1	<input type="radio"/>
2	<input type="radio"/>
3	<input checked="" type="radio"/>
4	<input type="radio"/>
5	<input type="radio"/>
Clear selection	
Performance of our softsoftware	
1	<input type="radio"/>
2	<input checked="" type="radio"/>
3	<input type="radio"/>
4	<input type="radio"/>
5	<input type="radio"/>
Clear selection	

Figure 98– create table astronomer

Overall rating

1

2

3

4

5

[Clear selection](#)

Suggestions for improvement

Make the system more user friendly

Feeback

Very good

[Submit](#) [Clear form](#)

Never submit passwords through Google Forms.

 This content is neither created nor endorsed by Google.
[Report Abuse](#) - [Terms of Service](#) - [Privacy Policy](#)

Figure 99– create table astronomer

Response 4

E_SPACE SOLUTION

aashikmahroof123@gmail.com

[Switch accounts](#)

Not shared

Draft saved

* Indicates required question

Your name *

Ahammed zacky

Email address *

ahammedzacky@gmail.com

Date *

Date

30/08/2024

Figure 100– create table astronomer

How user friendly is our software

1

2

3

4

5

Clear selection

Performance of our softsoftware

1

2

3

4

5

Clear selection

!

Figure 101

Overall rating

1

2

3

4

5

[Clear selection](#)

Suggestions for improvement

Improve more features

Feeback

Excellent

Figure 102– create table astronomer

Reponse 5

E_SPACE SOLUTION

aashikmahroof123@gmail.com
Switch accounts

✉ Not shared
Cloud Draft saved

* Indicates required question

Your name *

Mohammed yoousf

Email address *

mohammedyoousf12@gmail.com

Date *

Date
27/08/2024 ▾

0

Figure 103– create table astronomer

How user friendly is our software

1

2

3

4

5

Clear selection

Performance of our softsoftware

1

2

3

4

5

Clear selection

Figure 104—create table astronomer

Overall rating

1

2

3

4

5

[Clear selection](#)

Suggestions for improvement

Ui improvements

Feedback

Easy for use

 [Submit](#) [Clear form](#)

Figure 105– create table astronomer

Reponse 6

E_SPACE SOLUTION

aashikmahroof123@gmail.com
[Switch accounts](#)

 Not shared

 Draft saved

* Indicates required question

Your name *

Apsara

Email address *

apsarahndit@gmail.com

Date *

Date

29/08/2024 

Figure 106– create table astronomer

How user friendly is our software

1

2

3

4

5

[Clear selection](#)

Performance of our softsoftware

1

2

3

4

5

[Clear selection](#)

!

Figure 107 – create table astronomer

Overall rating

1

2

3

4

5

[Clear selection](#)

Suggestions for improvement

Due to security weakness use strong password or one time password

Feeback

Increase security to protect unauthorised acces

[Submit](#) [Clear form](#)

 Never submit passwords through Google Forms.

Figure 108 – create table astronomer

Reponse 7

E_SPACE SOLUTION

aashikmahroof123@gmail.com

[Switch accounts](#)

 Not shared

 Draft saved

* Indicates required question

Your name *

Kavindu herath

Email address *

kavinduherath@gmail.com

Date *

Date

28/09/2024

Figure 109 – create table astronomer

How user friendly is our software

1

2

3

4

5

Clear selection

Performance of our softsoftware

1

2

3

4

5

Clear selection

Figure 110 – create table astronomer

Overall rating

1

2

3

4

5

[Clear selection](#)

Suggestions for improvement

Need more attractive interfaces

Feeback

Good

[Submit](#) [Clear form](#)

Never submit passwords through Google Forms.

This content is neither created nor endorsed by Google. [R](#)
[Abuse](#) - [Terms of Service](#) - [Privacy Policy](#)

Figure 111 – create table astronomer

Overall feedback response

Based on the facts shown above, I can determine whether the system I designed is successful or not successful.

- ❖ Overall, the response I've gotten has been quite positive and straightforward to implement.

Feedback

7 responses



Figure 112 – create table astronomer

- ❖ The majority of the comments I've received indicates that program performance is sufficient

Performance of our software?

Copy

7 responses

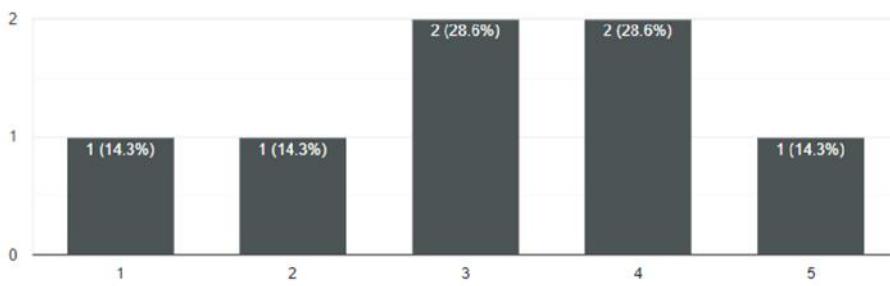


Figure 113 – create table astronomer

- ❖ Based on the overall feedback, I can conclude that the system is user friendly

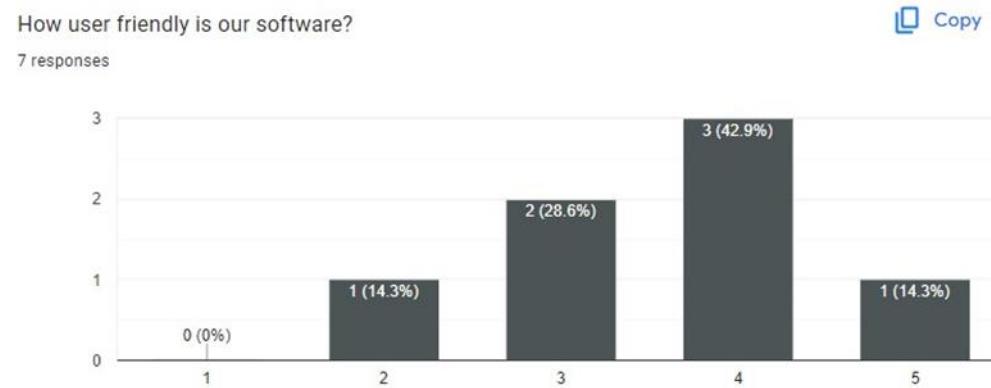


Figure 114 – create table astronomer

- ❖ The majority of people believe the system is generally acceptable

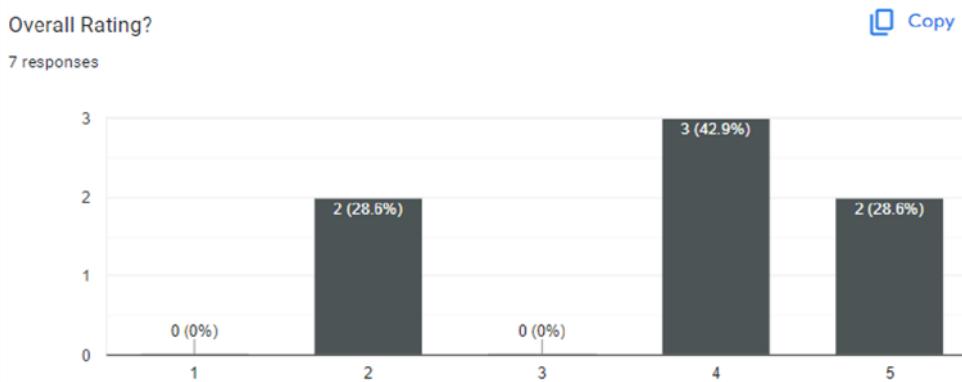


Figure 115

❖ Future improvement suggestions



Figure 116 – create table astronomer

Regarding the aforementioned ideas for improvement, the majority of them indicated that changes were needed. Such include increasing security, making the system more user friendly, and modernizing the UI. These recommendations were extremely beneficial to the author's investigation of the system in order to enhance it.

Activity 4

Introduction to the system

E-Space Solutions is developing the Mars Colonization Database Management System (MCDMS) to support their mission to colonize Mars by 2040. The system will manage data of volunteers, space crafts (E-Jets), pilots, trips to Mars, house assignments, and job assignments for colonists to ensure the project's success.

Overall, the Mars Colonization Database Management System is a comprehensive solution that will serve as the backbone of E-Space Solutions' mission to colonize Mars, ensuring that all data is managed efficiently and accurately to support the success of the project.

Technical documentation

Mars Colonization Database Management System (MCDMS)

Introduction

Mars Colonization Database Management System (MCDMS) version 1.0

Description:

The Mars Colonization Database Management System handle system is a piece of software designed to handle all of the system's records. Insert, update, delete, and search are the four primary functions available here.

User requirements of e-space solution

- ☒ The database should be able to store information about colonist, dependent, e-jet, trip, house and Astronomer.
- ☒ The database should allow for the creation of multiple passengers for a single trip as well as each house is assigned to a colonist.

- ☒ The database should have security measures in place to protect sensitive information.

- ☒ The database should have the ability to back up data regularly to prevent loss of information.
- ☒ The database should be user-friendly and easy to navigate.

- ☒ The database should have the ability to generate reports on the information such as Jet Detail report, Trip Detail report with passengers, Colonists details with dependents & etc. as needed.

- ☒ The database should maintain accurate information about trips and passengers details.

- ☒ The database should have an efficient data retrieval and storage method.

- The database should be able to store information about each colonist , including a name and any relevant details as well as The pilots can check the data of the system according to the space craft trip information with the passenger details
- The database should have high security in order to protect the colonists details
- When the database ongoing the Colony Superintendent will generate Colony report with House details & colonists, Job reports of the colonists & etc.
- The system should be able to create, add, edit and delete information using the system.

System requirements for of e-space solution

Software requirements

Operating System,
 Microsoft Data Access Components,
 SQL Server 2014 Management Studio
 Oracle
 Visual studio code 2019, pycham

Hardware Requirements

Processor: Intel Core (TM) i7-8550U CPU Processor 8 Cores 16 Threads
 (16M Cache, up to 5.10 GHz)

RAM: Corsair Vengeance LPX 32GB DDR4 SDRAM (2 x 16GB)
 3200MHz Desktop Memory

Storage: Samsung 970 EVO Plus 1TB NVMe SSD

Network: 1Gbps Ethernet LAN Card and Wi-Fi

Backup and Recovery: Synology DS420j 4-Bay NAS Enclosure with Seagate Iron Wolf 8TB NAS Internal Hard Drive.

System type: 64-bit operating system, x64 based processor

Other hardware: printers/scanners, Ups

USER GUIDE

Introduction

This user guide explains how to use the database system created for E-Space Solutions. The system is intended to manage data about colonist, dependent, E_jet, house, trip, job. The user guide assumes that the user is familiar with databases and SQL. Users must enter their login information. The administrator provides the username and password. Users must first select the appropriate table from the interface before inserting data into the database. Users can delete record by selecting it and then confirming the deletion. Users must be cautious when deleting data because it cannot be recovered once it has been deleted. After selecting the table, users can select the record to be updated and modify the fields as needed. This user guide provides basic instructions for using the E-Space Solutions database system. Users are advised to use caution when performing database operations and to ensure that the data entered is valid and adheres to the data type constraints.

Login page for E_space solution

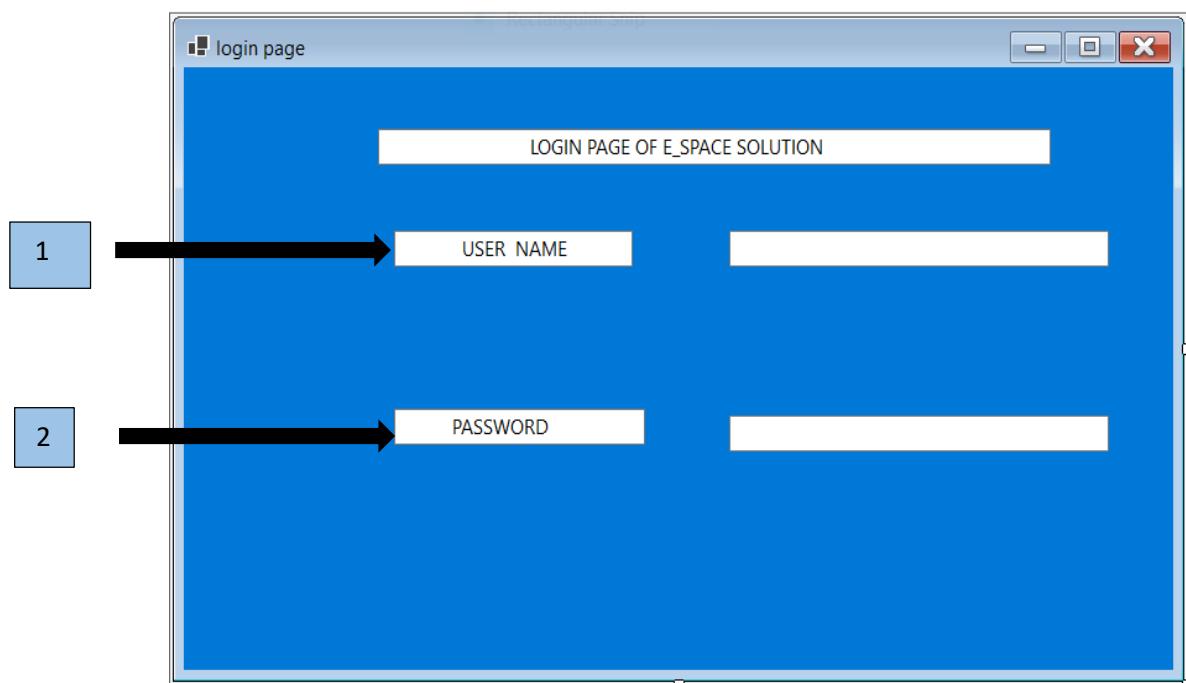


Figure 117

1. Enter User Name
2. Enter Password

Home page for E_space solution

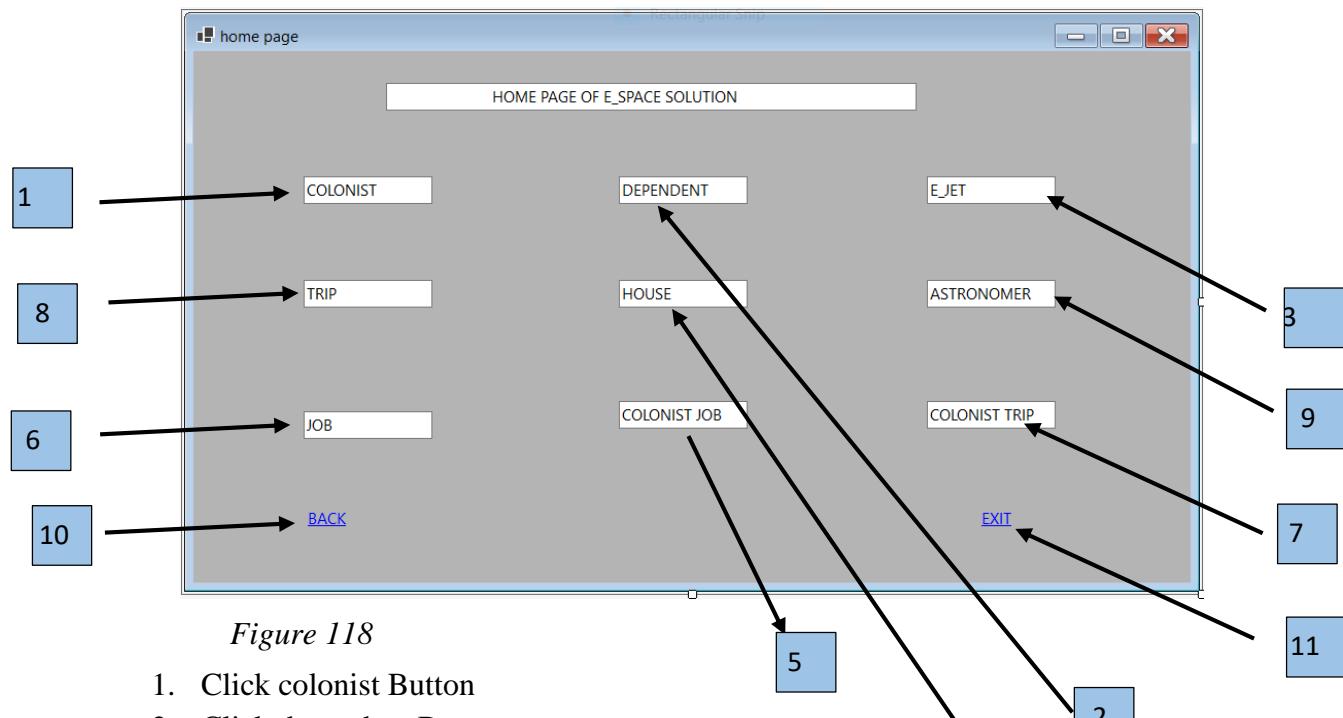
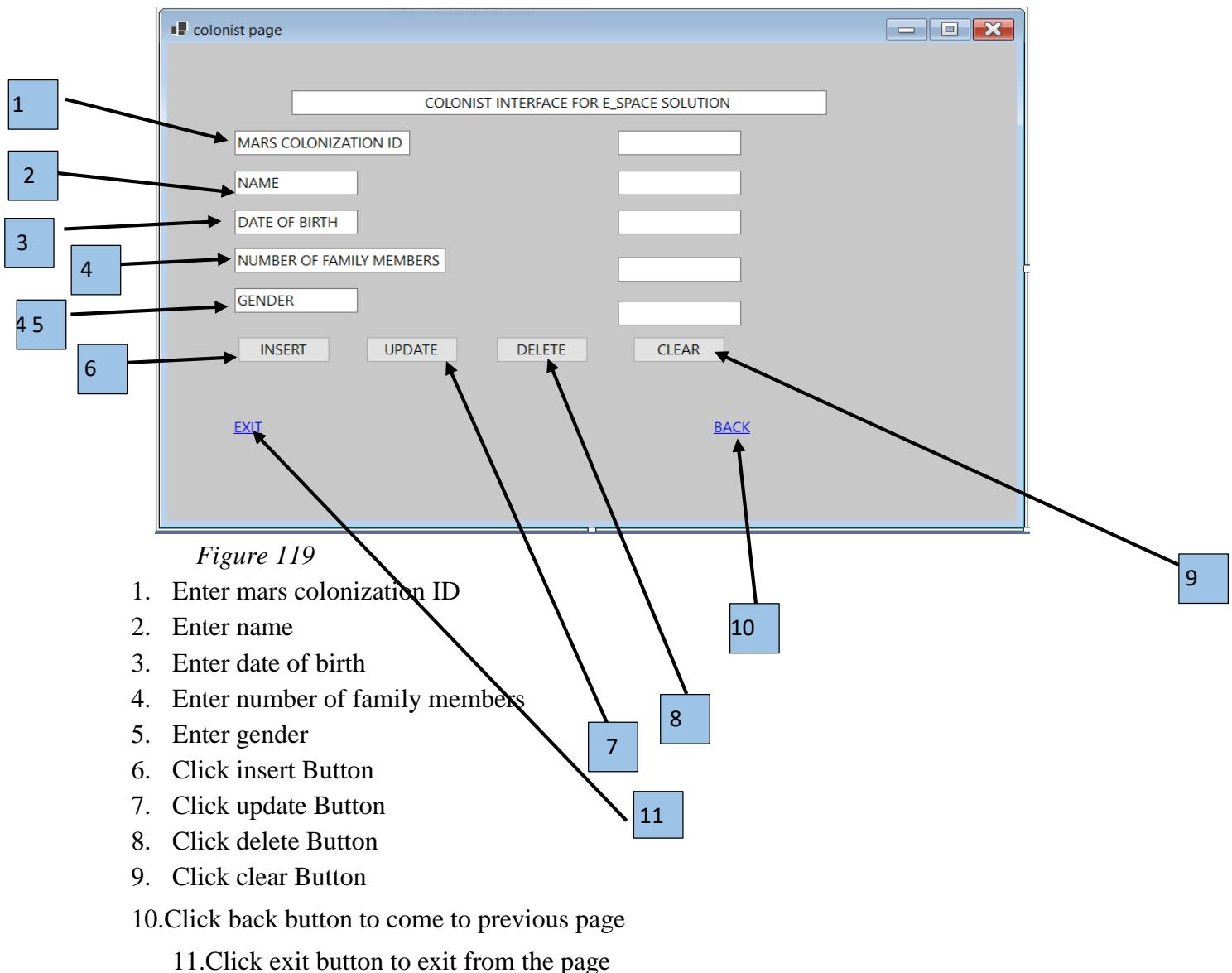


Figure 118

1. Click colonist Button
2. Click dependent Button
3. Click e_jet Button
4. Click house Button
5. Click colonist job Button
6. Click job Button
7. Click colonist trip Button
8. Click trip Button
9. Click astronomer Button
10. Click back button to come to previous page
11. Click exit button to exit from the page

Colonist interface for E_space solution



Dependent interface for E_space solution

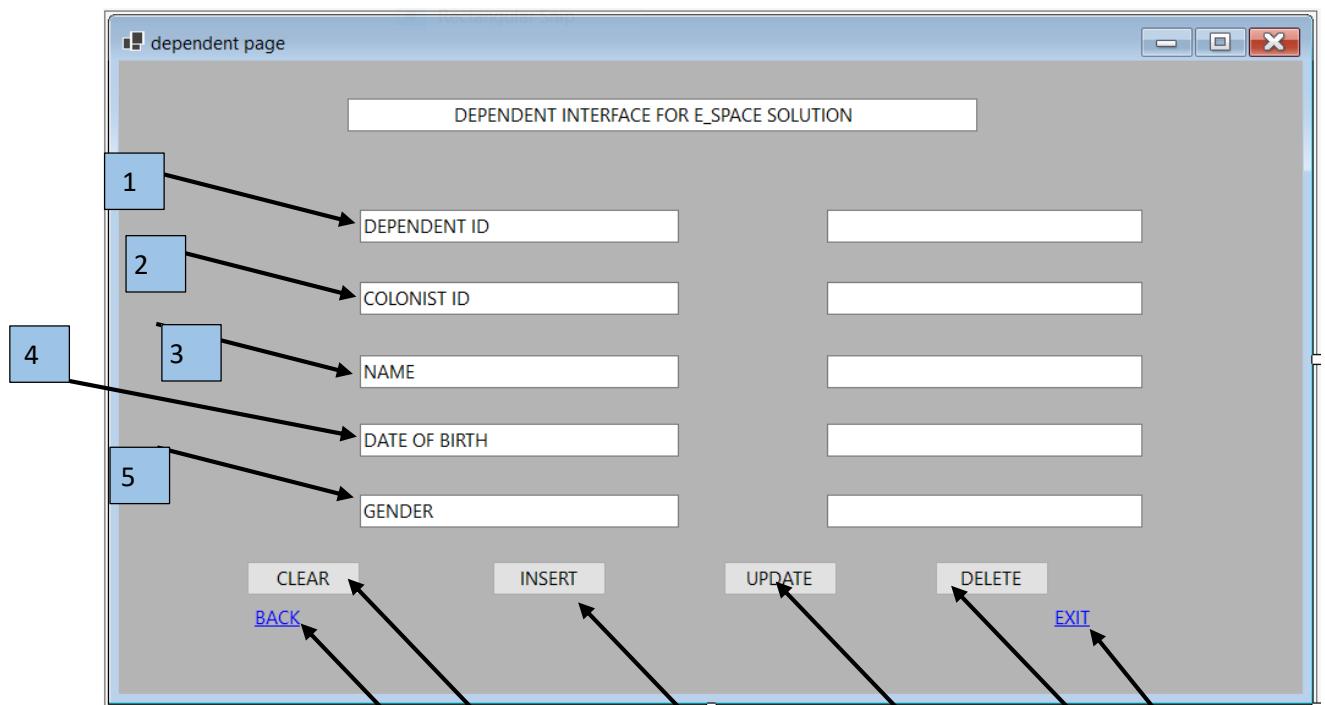


Figure 120 – create table astronomer

1. Enter dependent ID
2. Enter Colonist ID
3. Enter name
4. Enter date of birth
5. Enter gender
6. Click insert Button
7. Click update Button
8. Click delete Button
9. Click clear Button
10. Click back button to come to previous page
11. Click exit button to exit from the page

E_jet interface for E_space solution

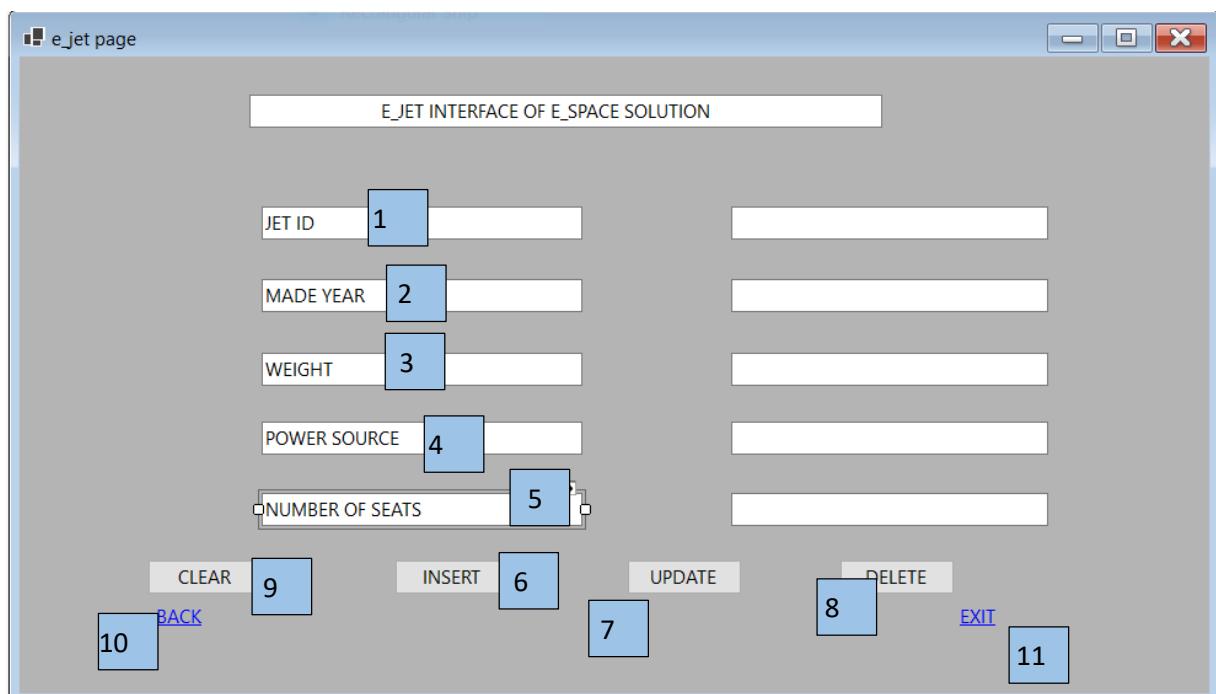


Figure 121—create table astronomer

- 1.Enter jet ID
- 2.Enter made year
- 3.Enter weight
- 4.Enter power source
- 5.Enter number of seats
- 6.Click insert Button
- 7.Click update Button
- 8.Click delete Button
- 9.Click clear Button
- 10.Click back button to come to previous page
- 11.Click exit button to exit from the page

Trip interface for E_space solution

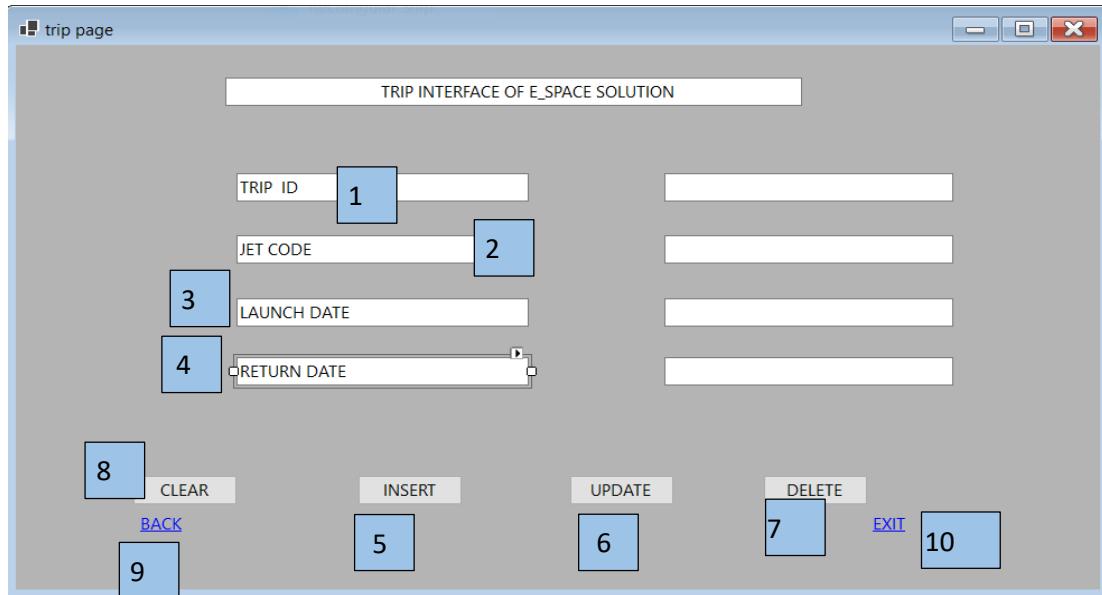


Figure 122 – create table astronomer

1. Enter trip ID
2. Enter jet code
3. Enter launch date
4. Enter return date
5. Click insert Button
6. Click update Button
7. Click delete Button
8. Click clear Button
9. Click back button to come to previous page
10. Click exit button to exit from the page

House interface for E_space solution

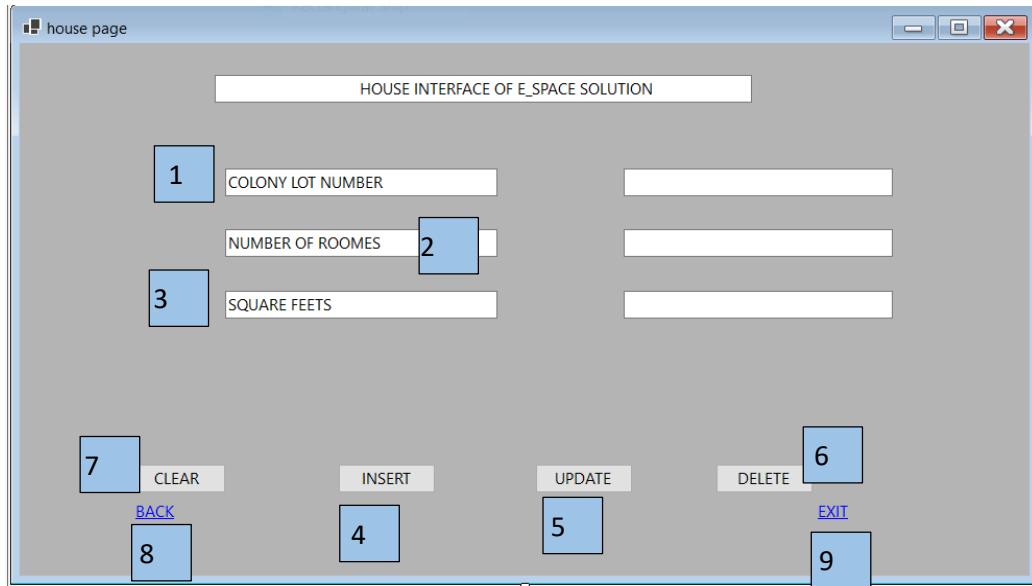


Figure 123 – create table astronomer

1. Enter colony lot number
2. Enter number of rooms
3. Enter square foots
4. Click insert Button
5. Click update Button
6. Click delete Button
7. Click clear Button
8. Click back button to come to previous page
9. Click exit button to exit from the page

Job interface for E_space solution

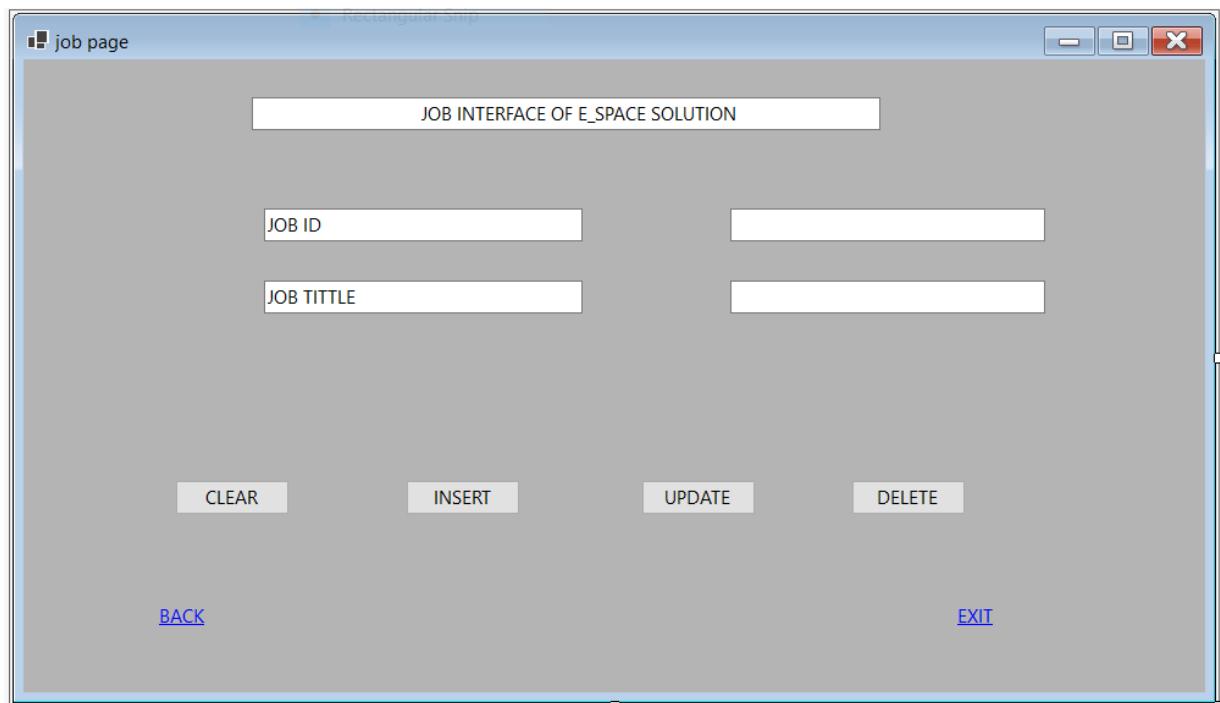


Figure 124 – create table astronomer

1. Enter jobID
2. Enter job title
3. Click insert Button
4. Click update Button
5. Click delete Button
6. Click clear Button
7. Click back button to come to previous page
8. Click exit button to exit from the page

UML Design for Scenario

Unified Modelling Language (UML) is a standard modelling language that consists of an integrated catalog of diagrams intended to help system architects, non-software developers, and business professionals define, visualize, build, and document IT systems artifacts. UML is a collection of the best engineering techniques that have been successfully applied to modelling large and complex systems. During object-oriented software development and throughout the software development process, UML is an essential component. The main purpose of graphics in UML is to describe the design of the software project. Project teams can come up with new ideas, communicate more effectively, and ensure that the software is architecturally sound when using UML.

UML Characteristics

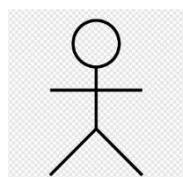
- It is a modeling language that has been generalized.
- It differs from other programming languages such as C++, Python, and others.
- It has a connection to object-oriented analysis and design.
- It is used to visualize the system's workflow.
- It is a graphical language for creating strong modeling objects.

Use Case Diagram

The main way that system and software requirements are expressed for a newly built software application is through a UML use case diagram. Use cases describe the desired action (what), not the precise process by which it should occur (how). When use cases are given, they can be represented textually or visually (use case diagram). The ability to create a system from the viewpoint of the end user is one of the main ideas behind use case modeling. It is a useful method for describing all externally observable system behavior and putting system behavior into the words of the user.

Use case diagram consists of 4 types

1. Actor



2. Use case

3. System
4. Package

USE CASE DIAGRAM FOR E-SPACE SOLUTIONS

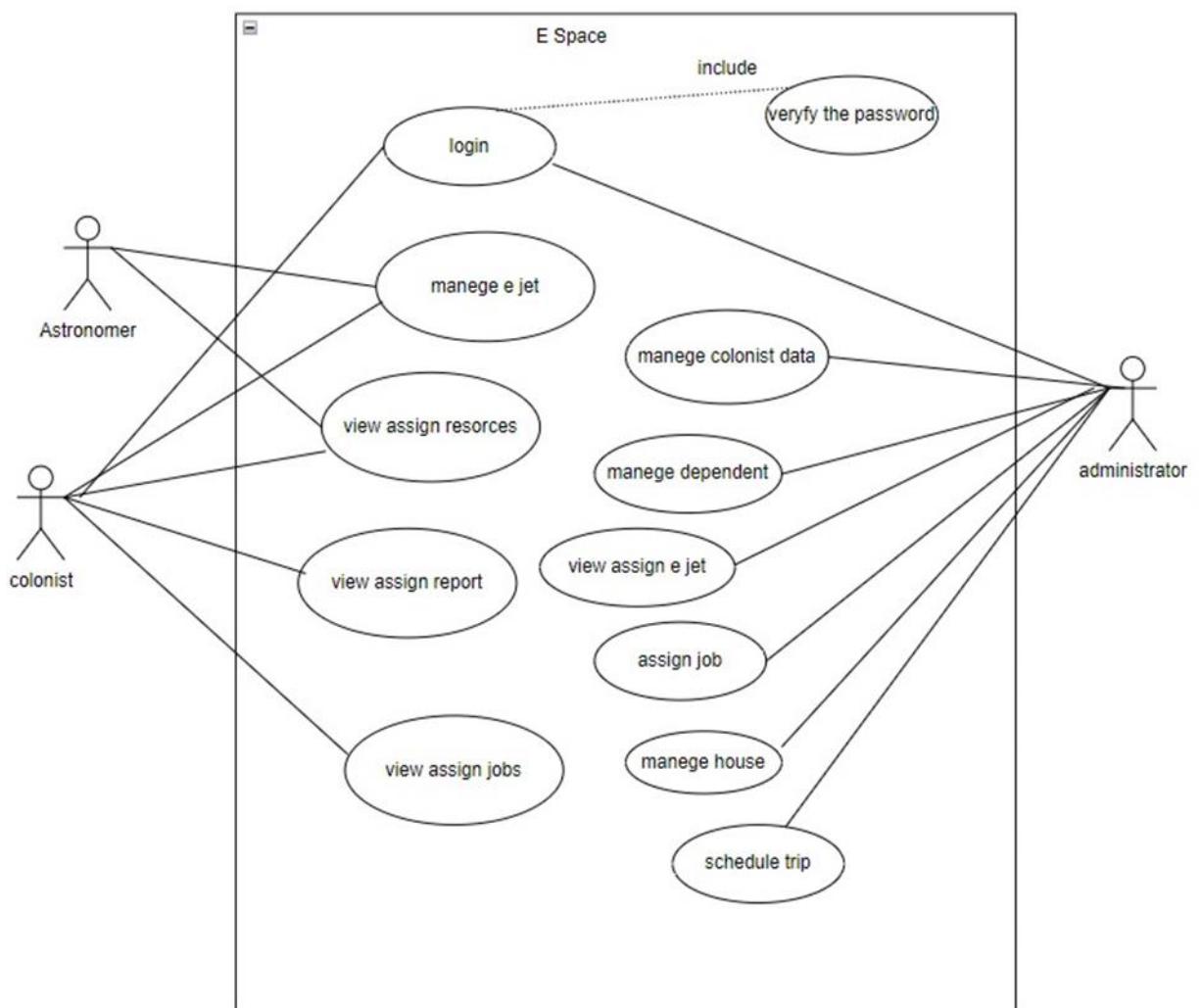


Figure 125

Class Diagram

In the Unified Modelling Language (UML), a class diagram is a kind of structural diagram that shows the organization and connections between classes in a system. It showcases the system's static structure by illuminating the classes, characteristics, actions, and methods as well as their relationships. Class diagrams are essential to object-oriented design because they aid in the visualization of system entities, attributes, and relationships. They also act as a development guide for software and give developers a clear grasp of the architecture of the system.

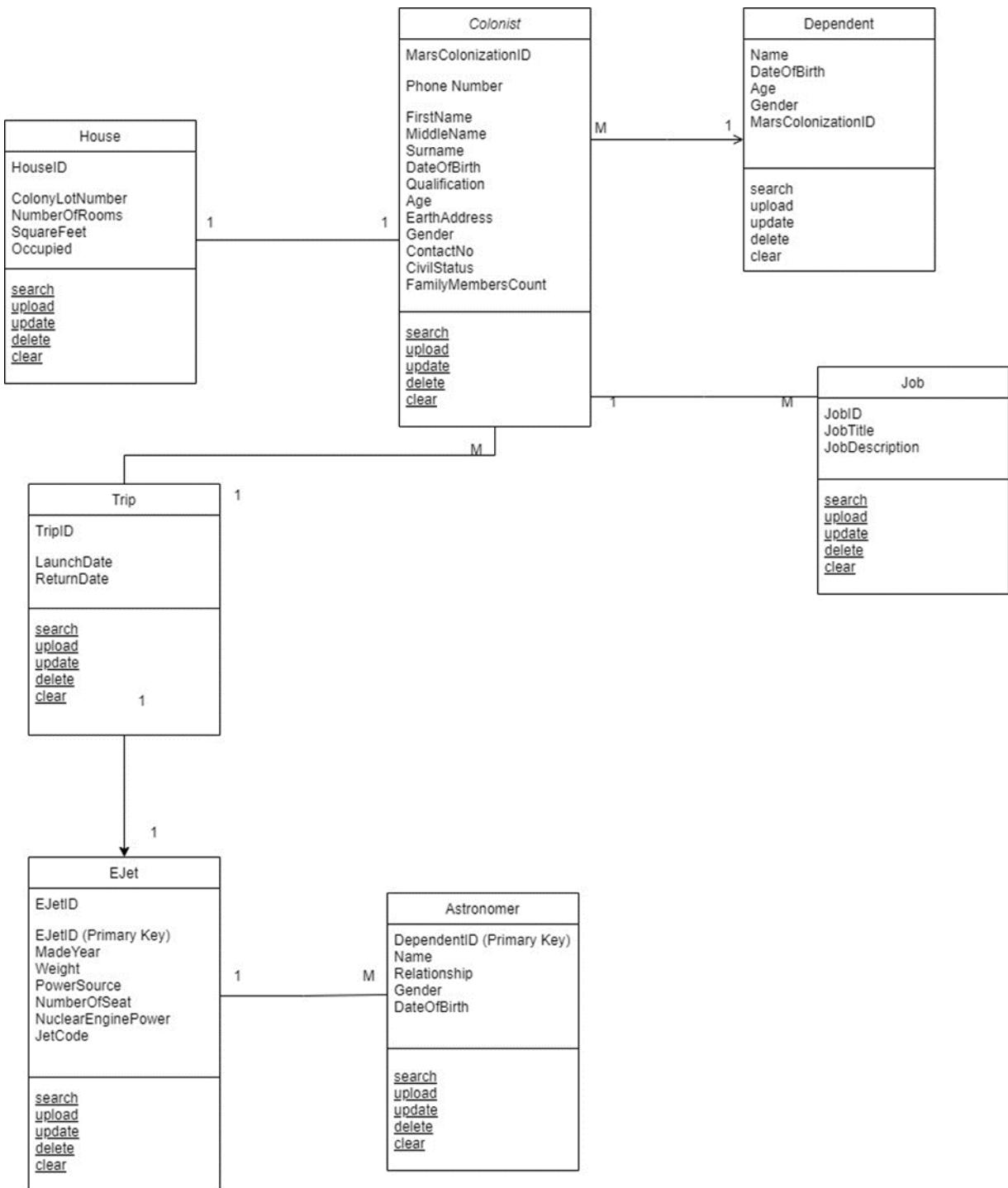


Figure 126

Flowchart

Flowcharts visually simplify complex processes using shapes like diamonds, ovals, and rectangles connected by arrows, aiding planning and analysis across various sectors. From basic sketches to detailed computer diagrams, they're widely used globally, referred to by different names like Process Flowchart or BPMN, and share similarities with UML Activity Diagrams and Data Flow Diagrams.

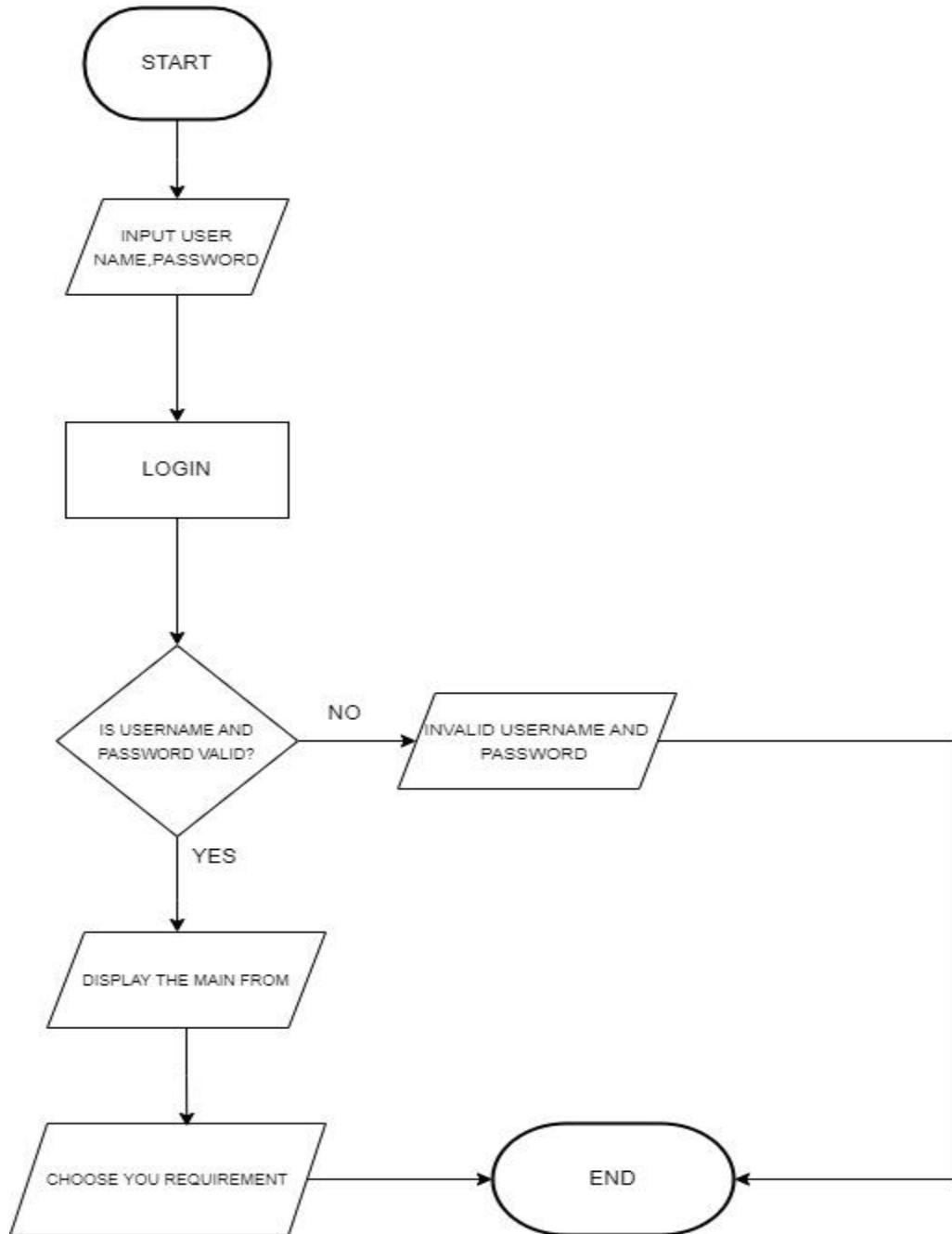


Figure 127 –

DFD-DATA FLOW DIAGRAM

LEVEL-0

Data Flow Diagram (DFD)

Level 0 DFD

Level 0 DFD provides a high-level overview of the system processes and data flow.

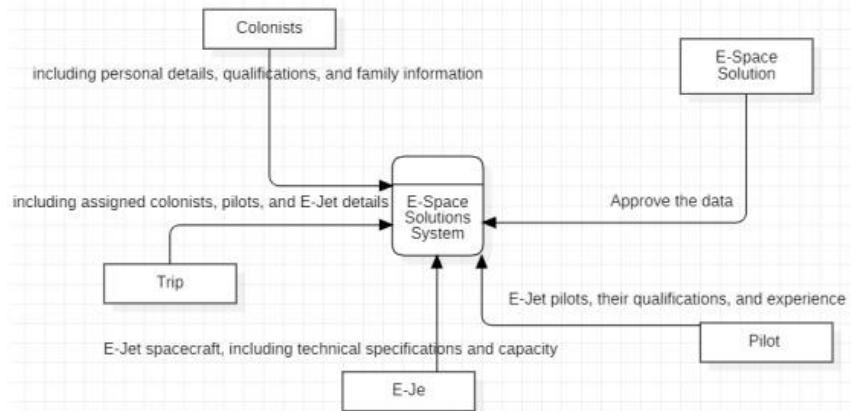


Figure 128 –

LEVEL-01

Level 1 DFD

Level 1 DFD breaks down the main processes into sub-processes for detailed understanding.

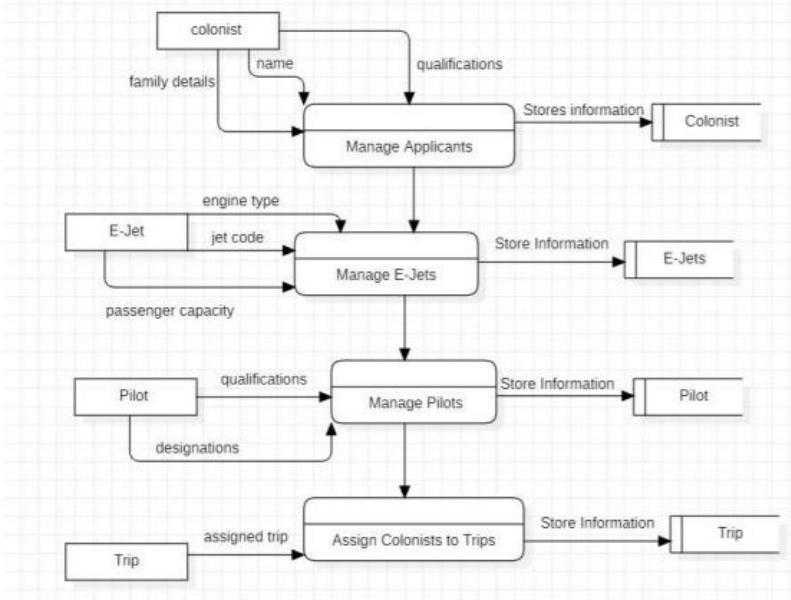


Figure 129

FUTURE ENHANCEMENT SUGGESTIONS FOR E-SPACE SOLUTION

Even though I worked hard to make the system as perfect as possible, there is always something to improve. There are always improvements that can be made to improve the system's effectiveness, from bad to good features. While the current system has basic reporting capabilities, adding more advanced reporting and analytics features could be useful for the company. This could include the ability to generate custom reports based on specific criteria, as well as visualizations such as graphs and charts to better understand colonist data. The login system in this database system has been improved. As in the current system, go to the SQL server and enter the username and password in the login table. This takes more time and can only be done by a developer or a trained professional. However, it would be ideal if we could manage the login table directly from the system. This allows an authorized user to add new usernames and passwords to the table directly from the system. This allows us to easily manage who has access to the database system. Improve system interfaces by adding more style and background images and making interfaces more suitable. We can also rearrange textboxes and buttons to make it more relevant. This will improve the system's appearance and add new features. As a result, more prefer will prefer and recommend our system to others. There are various interfaces for accessing various tables in the database. However, it would be very useful if we could access all of the tables from the same location. It would be very effective and increase the value of the system if we could access any table by selecting the table name. It will also be simple for the user. This can be done as a future development.

It would be advantageous to develop a mobile app to improve the efficiency and accessibility of a database management system for e-space solution .The software would allow changes and obtain information while on the go using their mobile devices. The system should also have collaboration tools that allow numerous users to work on the database at the same time and get real-time updates.

The protection of sensitive and private data from unauthorized access and hackers is achieved through the use of data encryption technologies, while user access control ensures that only authorized users have access to system features using unique login credentials. Backup and restoration measures are put in place to provide redundancy and safeguard data in case of data loss or system failures. Audit trails are used to track changes to the system and its data, while data validation and verification ensure the completeness and quality of data. Automated alerts are set up to notify users of security concerns, system faults, or suspicious activities, and regular vulnerability assessments are conducted to assess the security vulnerabilities of the system and implement necessary mitigation measures.

These future enhancement suggestions can help to ensure the system's long-term effectiveness and provide additional benefits to users.

Conclusion

In conclusion for this system that was as build for E_SPACE SOLUTION as a developer I have implemented the database by using SQL to be effective and the interface to be user-friendly by using VISUAL STUDIO, I have designed the system with proper security methods and new features so that the production can take advantage of it, if any future improvement required.

Gantt Chart

ACTIVITY NO	JUNE	JULY	AUGUEST	SEPTEMBER
ACTIVITY NO 1				
ACTIVITY NO 2				
ACTIVITY NO 3				
ACTIVITY NO 4				

Figure 130

References

References

- atzeni, p., 2009. *object relational database*. [Online] Available at: https://www.researchgate.net/figure/The-Object-Relational-model_fig14_220360554 [Accessed 10 08 2024].
- Beschokov, M., 2024. *firewall*. [Online] Available at: <https://www.wallarm.com/what/the-concept-of-a-firewall> [Accessed 12 08 2024].
- coach, I., 2021. *database design and development*. [Online] Available at: <https://www.learnerscoach.co.ke/database-design-and-development-course/amp/> [Accessed 01 08 2024].
- codersaty, 2021. *entity*. [Online] Available at: <https://www.geeksforgeeks.org/difference-between-entity-entity-set-and-entity-type/> [Accessed 03 08 2024].
- Everest, G. C., 1973. *flat file database*. [Online] Available at: https://www.researchgate.net/figure/Example-of-a-Flat-File_fig1_291448084 [Accessed 10 08 2024].
- glen, s., 2021. *primary key attributes*. [Online] Available at: <https://www.datasciencecentral.com/key-attributes-in-er-diagrams/> [Accessed 03 08 2024].
- Gupta, N., 2024. *a symmetric key encryption*. [Online] Available at: <https://www.ssl2buy.com/wiki/symmetric-vs-asymmetric-encryption-what-are-differences> [Accessed 27 08 2024].
- itskawal2000, 2022. *hierarchical database*. [Online] Available at: <https://www.geeksforgeeks.org/difference-between-hierarchical-and-network-data-model/> [Accessed 10 08 2024].
- KUMARI, N., 2024. *ER model*. [Online] Available at: <https://www.linkedin.com/pulse/er-model-nishi-kumari-t5uvf> [Accessed 04 08 2024].

- mangalgaishwarya2, 2024. *attributes.* [Online]
Available at: https://www.geeksforgeeks.org/types-of-attributes-in-er-model/?ref=header_outind
[Accessed 03 08 2024].
- padmalosani, 2023. *digital signature.* [Online]
Available at: <https://www.zoho.com/workdrive/digest/exploring-digital-signatures.html>
[Accessed 27 08 2024].
- schulte, o., n.d. *relational database.* [Online]
Available at: https://www.researchgate.net/figure/A-simple-relational-database-instance_fig1_266340568
[Accessed 10 08 2024].
- S, r. A., 2024. *ER Digram history.* [Online]
Available at: <https://www.simplilearn.com/tutorials/sql-tutorial/er-diagram-in-dbms#>
[Accessed 03 08 2024].
- S, R. A., 2024. *Normalization.* [Online]
Available at: <https://www.simplilearn.com/tutorials/sql-tutorial/what-is-normalization-in-sql>
[Accessed 05 08 2024].
- Vaughan, M. R. J., 2024. *database normalization.* [Online]
Available at: <https://www.techtarget.com/searchdatamanagement/definition/normalization>
[Accessed 05 08 2024].
- venkatachalam, m. a., 2024. *2024.* [Online]
Available at: <https://www.scaler.com/topics/anomalies-in-dbms/>
[Accessed 05 08 2024].