

Freebooks.pk

Free Notes • Past Papers • Guides for Students

2nd Year Computer Science — Punjab Board

CHAPTER 4

Data Integrity and Normalization

Key Concepts • Definitions • Short & Long Questions • MCQs • Diagrams • Exam Tips

© freebooks.pk — Original educational content by the Freebooks.pk Editorial Team



This chapter covers Data Integrity and Normalization from the 2nd Year (ICS Part-II) Computer Science syllabus of the Punjab Curriculum and Textbook Board (PTB/PCTB). It explains data integrity and its types, data anomalies, functional dependency and the process of normalization up to the third normal form. These notes are prepared by freebooks.pk.

For a database to be trustworthy, its data must be kept correct and free of harmful duplication. This chapter explains integrity rules and normalization, the techniques that ensure this.

Learning Objectives

- Define data integrity and its types.
- Explain entity and referential integrity.
- Describe the anomalies caused by redundancy.
- Explain functional dependency.
- Describe the purpose and process of normalization.
- State the first, second and third normal forms.

Key Concepts

Data Integrity

Data integrity means the correctness, accuracy and consistency of the data stored in a database. A database has integrity when its data obeys the rules that have been defined for it, so that it contains no wrong, missing or contradictory values. Integrity is enforced by integrity constraints, that is, rules built into the database, and it is essential because decisions are made using the data and incorrect data leads to wrong results.

Entity and Referential Integrity

Two important kinds of integrity are entity integrity and referential integrity. Entity integrity is the rule that the primary key of every table must have a value (it cannot be null) and that value must be unique, so that every record can be identified. Referential integrity is the rule that a foreign key value in one table must match an existing primary key value in the related table (or be null); this prevents a record from referring to another record that does not exist, keeping the links between tables valid.

Redundancy and Anomalies

When the same data is stored many times (redundancy), the database can suffer from anomalies, that is, problems that occur when data is changed. An insertion anomaly means some data cannot be added without adding other, unrelated data. An update anomaly means that changing one fact requires changing many copies, and if some are missed the data becomes inconsistent. A deletion anomaly means that deleting one record accidentally loses other useful data. Normalization is used to remove these anomalies.



Functional Dependency

A functional dependency describes how one field depends on another. If, for a given value of field A, there is always exactly one value of field B, then B is said to be functionally dependent on A, written $A \rightarrow B$. For example, a roll number determines a student's name, so Name is functionally dependent on RollNo. Understanding functional dependencies helps decide how to split tables correctly during normalization.

Normalization

Normalization is the process of organising the fields and tables of a database to reduce redundancy and remove anomalies. It is carried out in a series of steps called normal forms, in each of which the tables are refined a little further by splitting them into smaller, well-structured tables linked by keys. The aim is that each piece of data is stored in only one place, so the database becomes accurate, consistent and easy to maintain.

First and Second Normal Forms

A table is in first normal form (1NF) when it contains no repeating groups and every field holds only a single (atomic) value, so that the data is arranged as a simple table of rows and columns. A table is in second normal form (2NF) when it is already in 1NF and, in addition, every non-key field depends on the whole primary key and not on just part of it; this removes partial dependencies, which arise only when the primary key is composite (made of more than one field).

Third Normal Form

A table is in third normal form (3NF) when it is already in 2NF and, in addition, no non-key field depends on another non-key field; that is, there are no transitive dependencies. In practice this means that any field that really belongs to a different entity is moved out into its own table. A database whose tables are all in third normal form has very little redundancy and is generally free of the update, insertion and deletion anomalies, so 3NF is the usual goal of normalization.

Important Definitions

Term	Definition
Data integrity	The correctness, accuracy and consistency of stored data.
Entity integrity	The rule that a primary key must be unique and not null.
Referential integrity	The rule that a foreign key must match an existing primary key or be null.
Anomaly	A problem (insert, update or delete) caused by redundancy.
Functional dependency	When one field's value determines another ($A \rightarrow B$).
Normalization	Organising tables to remove redundancy and anomalies.
1NF	No repeating groups; every field holds a single value.



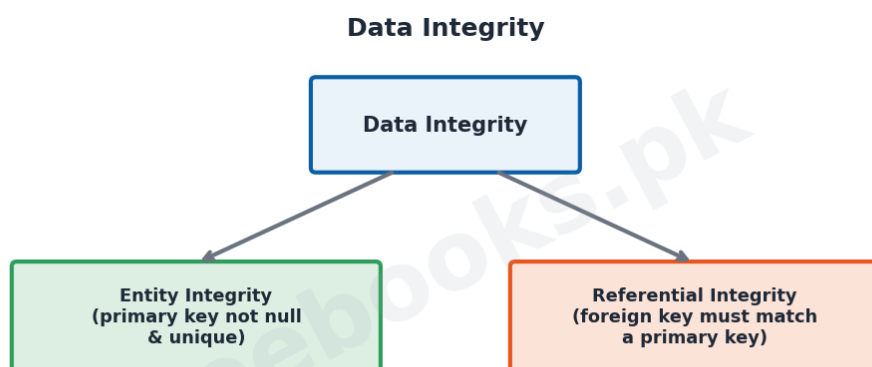
Term	Definition
3NF	In 2NF with no non-key field depending on another non-key field.

Key Facts & Rules

Item	Detail
Data integrity	Data is correct, accurate and consistent
Entity integrity	Primary key: unique and not null
Referential integrity	Foreign key matches a primary key (or null)
Anomalies	Insertion, update, deletion
Functional dependency	$A \rightarrow B$ (A determines B)
Normal forms	1NF $\rightarrow$ 2NF $\rightarrow$ 3NF

Diagrams & Illustrations

Data integrity: the two main kinds of data integrity: entity integrity (unique, non-null primary key) and referential integrity (valid foreign key).



Integrity rules keep the data correct and consistent

Data integrity

Normalization stages: the stages of normalization from unnormalized data through first, second and third normal forms.



Normalization: 1NF -> 2NF -> 3NF

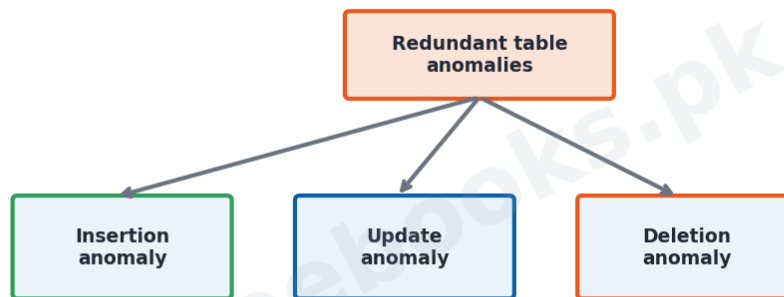


Normalization removes redundancy step by step

Normalization stages

Data anomalies: the insertion, update and deletion anomalies caused by redundant data, which normalization removes.

Anomalies in Un-normalized Data



Normalization avoids these anomalies

Data anomalies

Solved Examples & Practice

Entity integrity

A STUDENT table must not have two students with the same RollNo, and RollNo cannot be blank; this is entity integrity.

Referential integrity

A RESULT record with RollNo = 99 is not allowed if no student with RollNo 99 exists; this protects referential integrity.



Spot the anomaly

If a teacher's phone number is stored in every one of their class records, changing it means editing many rows; this is an update anomaly.

Functional dependency

In a STUDENT table, RollNo -> Name means the roll number determines the name.

Short Questions & Answers

Q1. What is data integrity?

Ans. The correctness, accuracy and consistency of the data stored in a database.

Q2. What is entity integrity?

Ans. The rule that a table's primary key must be unique and must not be null, so each record can be identified.

Q3. What is referential integrity?

Ans. The rule that a foreign key must match an existing primary key value in the related table (or be null).

Q4. Name the three data anomalies.

Ans. Insertion anomaly, update anomaly and deletion anomaly.

Q5. What is a functional dependency?

Ans. A relationship in which the value of one field determines the value of another, written A -> B.

Q6. What is normalization?

Ans. The process of organising tables to remove redundancy and the anomalies it causes.

Long Questions & Answers

Q1: Explain data integrity and describe entity integrity and referential integrity.

Data integrity means the correctness, accuracy and consistency of the data held in a database; a database is said to have integrity when all of its data obeys the rules laid down for it and contains no wrong, missing or contradictory values. Because organisations make important decisions using the data in their databases, and incorrect data would lead to wrong decisions, maintaining integrity is essential, and it is enforced by integrity constraints, which are rules built into the database that the DBMS checks whenever data is entered or changed. Two of the most important kinds of integrity are entity integrity and referential integrity. Entity integrity is concerned with the primary key of a table: the rule states that the primary key of every record must always have a value, that is, it can never be left null, and that this value must be unique, so that no two records share it. This guarantees that every record in the table can be uniquely identified; for example, in a STUDENT table no two students may have the same roll number and no student may be entered without one.



Referential integrity is concerned with the links between tables that are made by foreign keys: the rule states that the value of a foreign key in one table must either match an existing primary key value in the related table or be null. This prevents a record from referring to another record that does not exist; for instance, a RESULT record cannot contain a roll number for which there is no matching student in the STUDENT table. Together, entity integrity keeps each table internally correct, while referential integrity keeps the connections between tables valid, so that the whole database remains trustworthy.

Q2: Explain the anomalies caused by data redundancy and how normalization removes them.

When the same piece of data is stored in more than one place in a database, that is, when there is data redundancy, the database becomes prone to a group of problems called anomalies, which appear when data is inserted, changed or deleted. There are three kinds. An insertion anomaly occurs when it is impossible to add a piece of data without also adding other, unrelated data; for example, if the details of a course could only be stored as part of a student's record, then a new course could not be entered until at least one student had enrolled in it. An update anomaly occurs when a single fact is stored in many rows, so that changing it correctly requires changing every one of those rows; if even one copy is missed, the copies disagree and the data becomes inconsistent, for instance when a teacher's telephone number is repeated in every class record and only some are updated. A deletion anomaly occurs when deleting a record accidentally removes other useful information as well; for example, deleting the last student of a course might also delete the only stored details of that course. These anomalies all arise from redundancy, and the technique used to remove them is normalization. Normalization reorganises the fields and tables so that each fact is stored in only one place, by splitting large, redundant tables into smaller, well-structured tables that are linked together by keys. Once a database has been properly normalized, each item of data appears once, so it can be inserted, updated or deleted without any of these anomalies occurring, and the data stays consistent and correct.

Q3: Describe the process of normalization and explain the first, second and third normal forms.

Normalization is the step-by-step process of organising the tables of a database so as to remove redundancy and the anomalies that go with it, and it is carried out through a series of stages called normal forms, each of which imposes a stricter rule than the one before and refines the tables a little further. The first stage produces the first normal form (1NF). A table is in 1NF when it contains no repeating groups and every field holds only a single, indivisible (atomic) value, so that the data is arranged as a plain table of rows and columns with no lists or repeated columns; this gives the data a regular, tabular shape. The second stage produces the second normal form (2NF). A table is in 2NF when it is already in 1NF and, in addition, every non-key field depends on the whole of the primary key and not on just a part of it. This rule is only relevant when the primary key is a composite key, made up of more than one field, and it removes so-called partial dependencies by moving any field that depends on only part of the key into a separate table. The third stage produces the third normal form (3NF). A table is in 3NF when it is already in 2NF and, in addition, no non-key



field depends on another non-key field; such an indirect dependence is called a transitive dependency, and removing it means taking any field that really describes a different entity and placing it in its own table. When all the tables of a database are in third normal form, each piece of data is stored only once, redundancy is almost eliminated, and the update, insertion and deletion anomalies no longer occur, which is why reaching third normal form is the usual goal of normalization.

MCQs with Answers

1. The correctness and consistency of data is called:

(a) redundancy (b) data integrity (c) a query (d) an index

Correct Answer: (b) data integrity. Data integrity.

2. A primary key must be unique and:

(a) null (b) not null (c) repeated (d) empty

Correct Answer: (b) not null. Not null (entity integrity).

3. A foreign key must match a ___ or be null:

(a) field (b) primary key (c) view (d) index

Correct Answer: (b) primary key. A primary key (referential integrity).

4. A problem caused by redundancy when changing data is an:

(a) entity (b) anomaly (c) index (d) attribute

Correct Answer: (b) anomaly. An anomaly.

5. A $\rightarrow$ B means A ___ B:

(a) equals (b) determines (c) deletes (d) copies

Correct Answer: (b) determines. Determines (functional dependency).

6. Removing redundancy by reorganising tables is:

(a) indexing (b) normalization (c) hashing (d) backup

Correct Answer: (b) normalization. Normalization.

7. Every field holding a single value gives:

(a) 1NF (b) 2NF (c) 3NF (d) ONF

Correct Answer: (a) 1NF. First normal form (1NF).

8. Removing partial dependency gives:

(a) 1NF (b) 2NF (c) 3NF (d) BCNF

Correct Answer: (b) 2NF. Second normal form (2NF).

9. Removing transitive dependency gives:

(a) 1NF (b) 2NF (c) 3NF (d) ONF

Correct Answer: (c) 3NF. Third normal form (3NF).

10. The usual goal of normalization is:

(a) ONF (b) 1NF (c) 2NF (d) 3NF

Correct Answer: (d) 3NF. 3NF.



Quick Revision Summary

- Data integrity = correct, accurate, consistent data (enforced by constraints).
- Entity integrity: primary key unique + not null. Referential integrity: foreign key matches a primary key.
- Redundancy causes anomalies: insertion, update, deletion.
- Functional dependency $A \rightarrow B$: A determines B.
- Normalization removes redundancy by splitting tables (linked by keys).
- 1NF (atomic values), 2NF (no partial dependency), 3NF (no transitive dependency) = goal. Notes by freebooks.pk.

Exam Tips

- Define data integrity and its two main types.
- Explain entity vs referential integrity clearly.
- Name and explain the three anomalies.
- State a functional dependency ($A \rightarrow B$) with an example.
- Give the rule for each of 1NF, 2NF and 3NF.
- State that 3NF is the usual goal.

© freebooks.pk — DMCA-protected. May be shared or linked without changes. Original content by the Freebooks.pk Editorial Team.

