CHANDIGARH UNIVERSITY

Discover. Learn. Empower.

INSTITUTE : UIE
DEPARTMENT : APEX INSTITUTE OF TECHNOLOGY(CSE) -AIML

Bachelor of Engineering (Computer Science & Engineering)

Advanced Database Management

Er. Vishwa Deepak (E12867)

DISCOVER . LEARN . EMPOWER

Transaction Control

Course Objectives

CO Number	Course Objective	Level
CO1	Develop understanding the advancement in SQL	Apply

Course Outcome

CO Number	Course Outcome	Level
CO1	Describe and execute advanced level SQL queries	Apply

LECTURE OUTCOMES

❖Student will learn about the Transaction Control Language

1. Transaction processing
2. Transaction control language
3. Transaction control command : Commit
4. Summary

TRANSACTION

- Collection of operations that form a single logical unit of work are called transactions.
- In other words, A transaction is a unit of program execution that accesses and possibly updates various data items.
- A transaction is delimited by statements (or function calls) of the form begin transaction and end transaction. The transaction consists of all operations executed between the begin transaction and end transaction.
- A transaction groups SQL statements so that the changes performed by them, either all are **committed**, which means they are applied to the database, or all **rolled back**, which means they are undone/cancelled from the database.

TRANSACTION OPERATIONS

Let's understand the transaction concept using a simple bank application consisting of several accounts and a set of transactions that access and update those accounts. Transactions access data using two operations:

read(X):which transfers the data item X from the database to a variable, also called X, in a buffer in main memory belonging to the transaction that executed the read operation.

write(X): which transfers the value in the variable X in the main-memory buffer of the transaction that executed the write to the data item X in the database.

EXAMPLE

- Example: A transaction that transfers $50 from account A to account B.

read(A);

$A := A - 50;$

write(A);

read(B);

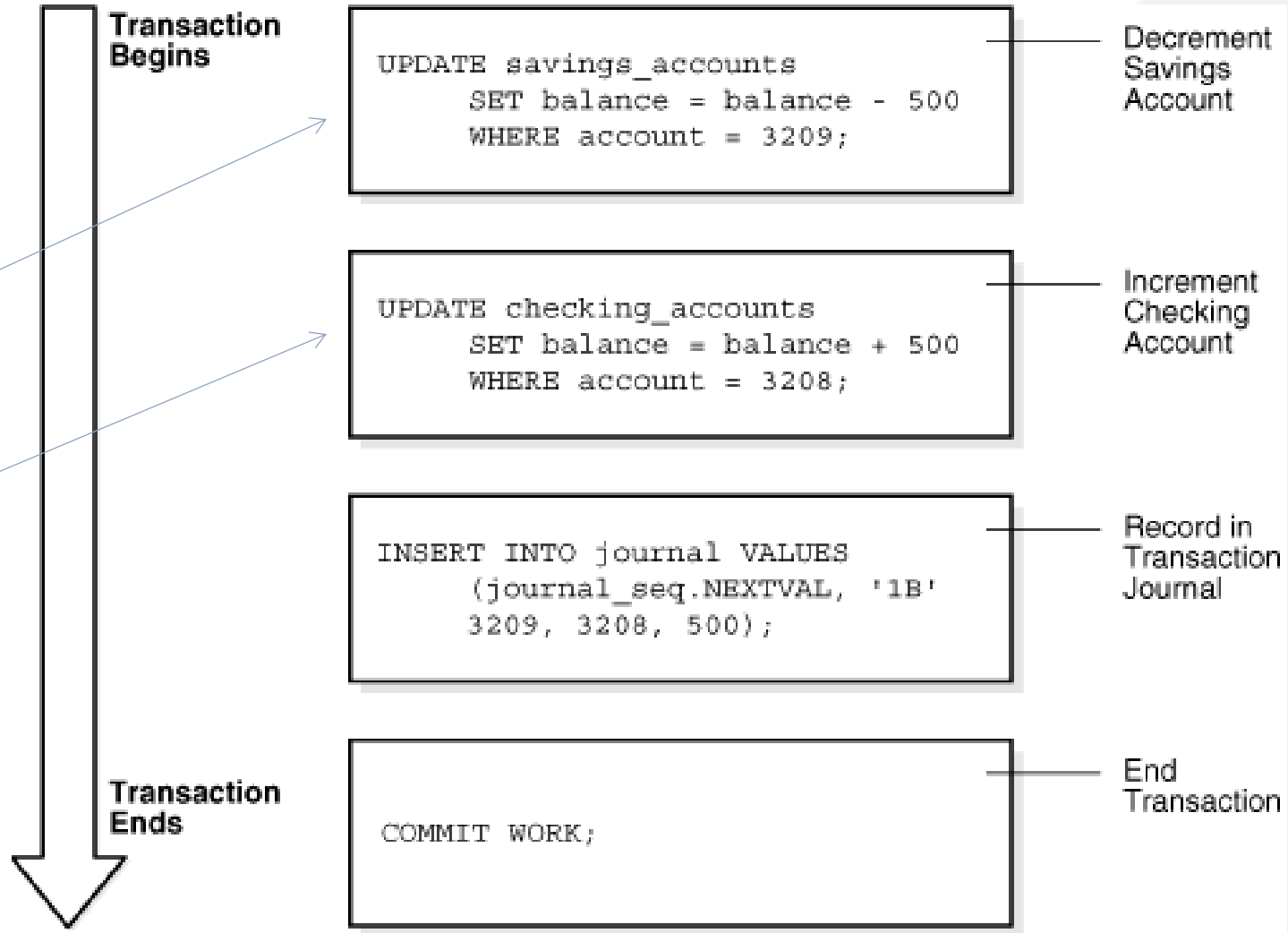
$B := B + 50;$

write(B).

TRANSACTION STRUCTURE

- A database transaction consists of one or more ***data manipulation language (DML)*** statements that together constitute an **atomic** change to the database

1. **read**(*A*)
2. *A* := *A* – 50
3. **write**(*A*)
4. **read**(*B*)
5. *B* := *B* + 50
6. **write**(*B*)

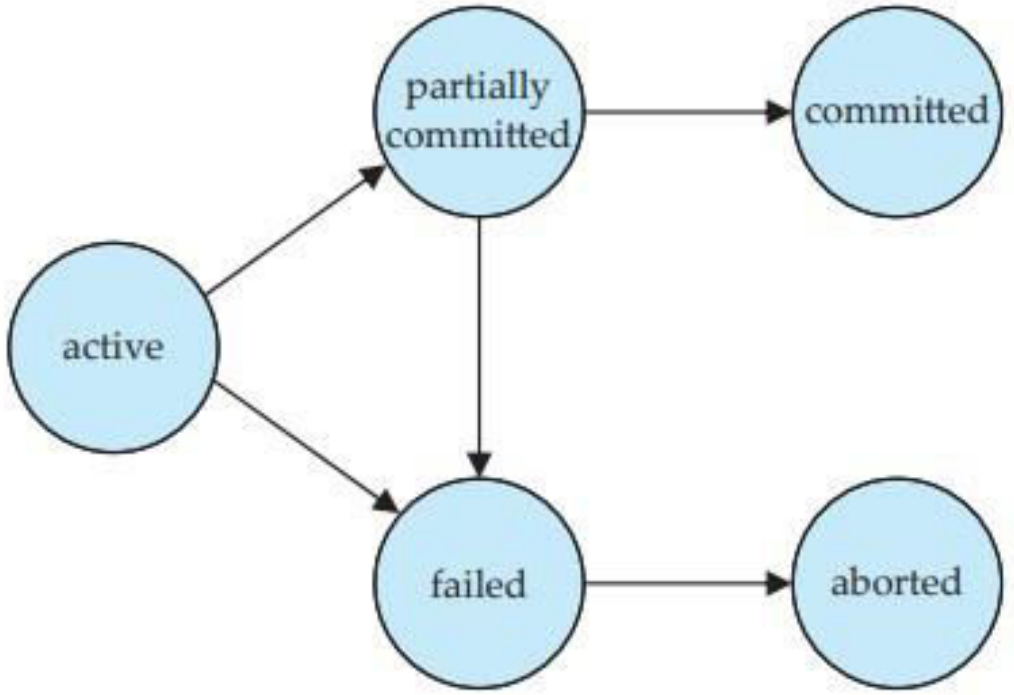


https://docs.oracle.com/database/121/CNCPT/img/GUID-ABBF8589-449F-44BA-946B-858569B2ECB6-default.gif

TRANSACTION PROPERTIES

- **Atomicity:** Either all operations of the transaction are reflected properly in the database, or none are.
- **Consistency:** Execution of a transaction in isolation (that is, with no other transaction executing concurrently) preserves the consistency of the database.
- **Isolation:** Even though multiple transactions may execute concurrently, the system guarantees that, for every pair of transactions Ti and Tj , it appears to Ti that either Tj finished execution before Ti started or Tj started execution after Ti finished. Thus, each transaction is unaware of other transactions executing concurrently in the system.
- **Durability:** After a transaction completes successfully, the changes it has made to the database persist, even if there are system failures.

STATE DIAGRAM OF TRANSACTION



DATABASE LANGUAGES

Data definition language (DDL)	Data manipulation language (DML)	Data control language(DCL)	Transaction control language (TCL)
CREATE:	SELECT:	GRANT	COMMIT
ALTER:	INSERT	REVOKE	ROLLBACK
DROP:	UPDATE		SAVEPOINT
RENAME:	DELETE		

TRANSACTION CONTROL LANGUAGE

- Transaction control language (TCL) manages the transactions within a database. Transactions group a set of related tasks into a single, executable task.
- These are used to manage the changes made by DML-statements.
- All the tasks must succeed in order for the transaction to work.

TCL commands:

❖**COMMIT:** Carries out a transaction

❖**ROLLBACK:** Restores a transaction if any tasks fail to execute

❖**SAVEPOINT:** Sets a point in a transaction to save

C O M M I T

- Commit command is used to permanently save any transaction into the database.
- It ends the current transaction and makes permanent changes during the transaction.
- **Syntax:**

  Commit;

Example:

DELETE FROM Student

WHERE Age = 25;

COMMIT;

COMMIT CONTINUE...

- Commit command is the last statement of the transaction to save the changes permanently made by the transaction.
- A transaction that fails to successfully complete its execution will have an **abort** instruction as the last statement
- The COMMIT command saves all the transactions to the database since the last COMMIT or ROLLBACK command.

SUMMARY

- Covered the transaction and its properties for the successful transaction.
- Explained the transactional control language command commit.

HOME WORK

- What are the ACID properties?
- What Commit command does?

REFERENCES

- **Reference book:** Database Systems concepts, Korth
- **Web References:**
- ORACLE https://docs.oracle.com/cd/E11882_01/server.112/e40540/transact.htm#CNCPT1115
- https://www.tutorialspoint.com/sql/sql-transactions.htm
- https://www.geeksforgeeks.org/sql-ddl-dml-tcl-dcl/