

5. Understanding Transactions

Definition and Importance

A transaction in databases is a sequence of SQL operations executed as a logical unit of work. Its importance lies in ensuring data integrity and consistency by adhering to the ACID properties:

- **Atomicity:** All operations within the transaction are completed, or none are executed.
 - **Consistency:** The database transitions from one valid state to another valid state.
 - **Isolation:** Concurrent transactions do not affect each other's results.
 - **Durability:** Committed changes persist even in case of system failures.
-

6. Rolling Back Transactions

The ROLLBACK command allows reversing a transaction if an error occurs. It is helpful in situations where the database must remain in a consistent state.

7. Understanding Record Locking Policies

Types of Locking

1. **Pessimistic Locking:** Locks the record until the transaction completes, preventing concurrent modifications.

```
START TRANSACTION;  
SELECT * FROM inventory WHERE inventory_id = 5 FOR UPDATE; //locking  
UPDATE inventory SET last_update = NOW() WHERE inventory_id = 5;  
COMMIT;
```

Leon_Ashley_Final_Report_Sakila :D

2. **Optimistic Locking:** This does not lock records but checks if the data changed before committing.

```
START TRANSACTION;
SELECT inventory_id, last_update FROM inventory WHERE inventory_id = 5; // without
locking
UPDATE inventory
SET last_update = NOW()
WHERE inventory_id = 5 AND last_update = '2024-03-14 19:00:00';

IF ROW_COUNT() = 0
THEN
    ROLLBACK;
ELSE
    COMMIT;
END IF;
```

This method is used in low-concurrency systems to improve performance.

8. Ensuring Data Integrity and Consistency

Common Data Integrity Issues in Sakila

- Lack of data validation and recordings of payments or rental.
- Deleting customers with active rentals.
- Missing foreign key constraints that don't connect the IDs.
- Duplicate information.

Implementing Foreign Keys = Example

```
ALTER TABLE rental
ADD CONSTRAINT fk_rental_customer FOREIGN KEY (customer_id) REFERENCES
customer(customer_id);
```

Implementing Triggers for Data Consistency = Example

Preventing the deletion of customers with active rentals:

(Chat GPT - EXAMPLE)

Leon_Ashley_Final_Report_Sakila :D

```
DELIMITER //
CREATE TRIGGER before_customer_delete
BEFORE DELETE ON customer
FOR EACH ROW
BEGIN
    IF (SELECT COUNT(*) FROM rental WHERE customer_id = OLD.customer_id) > 0 THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Cannot delete: Customer has active rentals';
    END IF;
END;
//
DELIMITER ;
```

This trigger prevents inconsistencies when deleting related data.

Research and document different SQL statements used to modify database content. (**INSERT, UPDATE, DELETE, ALTER, etc.**).

Inserting new data

The syntax for inserting new data in a table is to select it with "**INSERT INTO** *name_table*", and to insert the new row, you must respect the order of the columns.

For example, in this first syntax, we introduce values in every column and order.

Insert statement with values for all columns

```
INSERT INTO mytable
VALUES (value_or_expr, another_value_or_expr, ...),
       (value_or_expr_2, another_value_or_expr_2, ...),
       ...;
```

In contrast with the first one, we only introduce values in specified columns. After the "**INSERT TO** *name_table*", we add the columns in brackets, which makes a specified order you must respect, and place the new values.

Leon_Ashley_Final_Report_Sakila :D

Insert statement with specific columns

```
INSERT INTO mytable
(column, another_column, ...)
VALUES (value_or_expr, another_value_or_expr, ...),
       (value_or_expr_2, another_value_or_expr_2, ...),
       ...;
```

Example: (if it's a String, put it in between quotation marks "")

Example Insert statement with expressions

```
INSERT INTO boxoffice
(movie_id, rating, sales_in_millions)
VALUES (1, 9.9, 283742034 / 1000000);
```

Updating rows

Update is used to modify data on tables, which allows changing the data only in the rows that meet the condition.

Update statement with values

```
UPDATE mytable
SET column = value_or_expr,
    other_column = another_value_or_expr,
    ...
WHERE condition;
```

The syntax of this query is **UPDATE** *name_table*, **SET** *column = dato nuevo... etc.*, **WHERE** *condition*.

Example:

```
UPDATE Customers
SET ContactName = 'Alfred Schmidt', City= 'Frankfurt'
WHERE CustomerID = 1;
```

******Recommendation: First use SELECT to see which table we're changing***

Deleting rows

Leon_Ashley_Final_Report_Sakila :D

Delete allows erasing data in the rows that meet the condition.

Delete statement with condition

```
DELETE FROM mytable  
WHERE condition;
```

***If you don't put a condition, it will erase everything on the table.

Altering tables

It's used for adding, deleting, or modifying columns or constraints.

Adding columns

ADD ... DEFAULT (opcional)

Altering table to add new column(s)

```
ALTER TABLE mytable  
ADD column DataType OptionalTableConstraint  
      DEFAULT default_value;
```

Removing columns

DROP...

Altering table to remove column(s)

```
ALTER TABLE mytable  
DROP column_to_be_deleted;
```

Modify column

MODIFY COLUMN...

```
ALTER TABLE table_name  
MODIFY COLUMN column_name datatype;
```

Describe the functionalities of MySQL Workbench tools such as **SQL Editor, Schema Inspector, and Query Builder.**

Leon_Ashley_Final_Report_Sakila :D

SQL Editor

- Allows users to write, edit, and execute SQL queries.
- Keeps a history of executed queries for easy reference.
- It supports multiple tabs that work on different scripts at the same time.
- Allows import and export of databases

Schema Inspector

- Provides detailed metadata about database schemas, tables, columns, indexes, and foreign keys.
- Offers tools to check and repair table integrity.
- Supports visual representation of table relationships.

Query Builder

- A graphical interface for designing SQL queries without manual coding.
- Allows users to drag and drop tables to define connections.
- Automatically generates SQL code based on selections.

External references:

https://www.w3schools.com/MySQL/mysql_foreignkey.asp

<https://chatgpt.com/>

<https://dev.mysql.com/doc/refman/8.4/en/commit.html>

<https://dev.mysql.com/doc/refman/8.4/en/trigger-syntax.html>

<https://www.freecodecamp.org/news/how-to-use-mysql-transactions/>

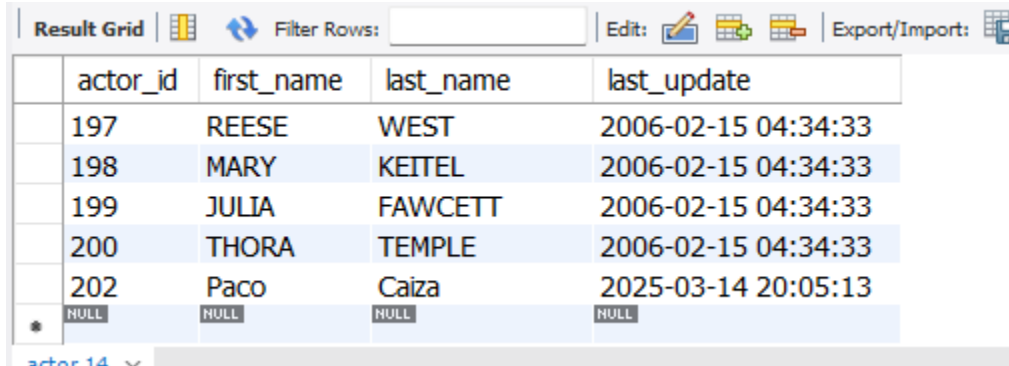
<https://www.datasunrise.com/knowledge-center/data-integrity-in-mysql/>

<https://github.com/basesdepatos/notes/tree/main/unit1>

SCREENSHOTS

Data Insertion, Deletion, and Update

- Using the actor table, insert a new record with fictitious data.

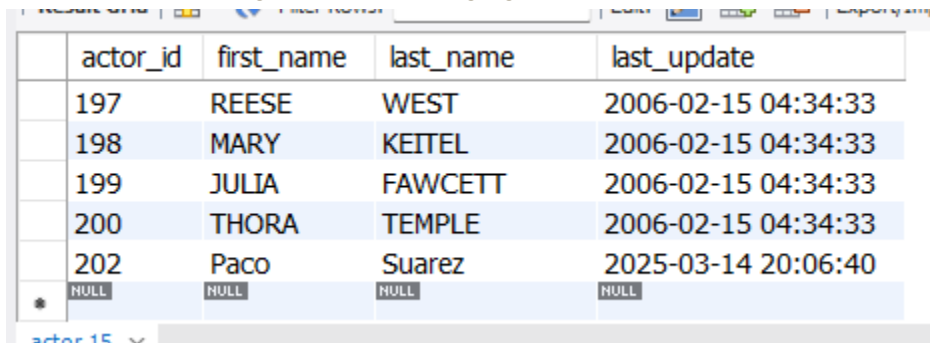


The screenshot shows a database interface with a 'Result Grid' and a table of actors. The table has columns: actor_id, first_name, last_name, and last_update. The data is as follows:

actor_id	first_name	last_name	last_update
197	REESE	WEST	2006-02-15 04:34:33
198	MARY	KEITEL	2006-02-15 04:34:33
199	JULIA	FAWCETT	2006-02-15 04:34:33
200	THORA	TEMPLE	2006-02-15 04:34:33
202	Paco	Caiza	2025-03-14 20:05:13
NULL	NULL	NULL	NULL

The interface includes a 'Filter Rows' field, an 'Edit' button, and an 'Export/Import' button. The table is titled 'actor 14'.

- Update an existing record by changing the last name of an actor.

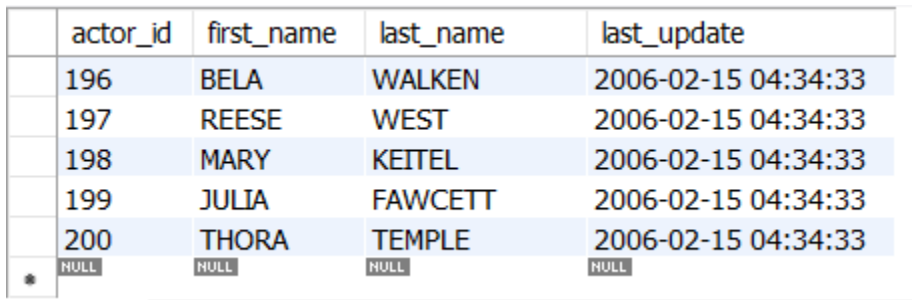


The screenshot shows a database interface with a 'Result Grid' and a table of actors. The table has columns: actor_id, first_name, last_name, and last_update. The data is as follows:

actor_id	first_name	last_name	last_update
197	REESE	WEST	2006-02-15 04:34:33
198	MARY	KEITEL	2006-02-15 04:34:33
199	JULIA	FAWCETT	2006-02-15 04:34:33
200	THORA	TEMPLE	2006-02-15 04:34:33
202	Paco	Suarez	2025-03-14 20:06:40
NULL	NULL	NULL	NULL

The interface includes a 'Filter Rows' field, an 'Edit' button, and an 'Export/Import' button. The table is titled 'actor 15'.

- Delete an actor from the table.



The screenshot shows a database interface with a 'Result Grid' and a table of actors. The table has columns: actor_id, first_name, last_name, and last_update. The data is as follows:

actor_id	first_name	last_name	last_update
196	BELA	WALKEN	2006-02-15 04:34:33
197	REESE	WEST	2006-02-15 04:34:33
198	MARY	KEITEL	2006-02-15 04:34:33
199	JULIA	FAWCETT	2006-02-15 04:34:33
200	THORA	TEMPLE	2006-02-15 04:34:33
NULL	NULL	NULL	NULL

The interface includes a 'Filter Rows' field, an 'Edit' button, and an 'Export/Import' button. The table is titled 'actor 16'.

Leon_Ashley_Final_Report_Sakila :D

Creating a Table from a Query Result

```
5 -- We created a table with those results
6 • CREATE TABLE recent_films AS
7 SELECT * FROM film WHERE release_year>2005;
```

film_id	title	description	release_year	language_id	original_language_id	rental_duration	rental_rate	length	replacement_cost	rating	special_features
1	ACADEMY DINOSAUR	A Epic Drama of a Feminist And a Ma...	2006	1	NULL	6	0.99	86	20.99	PG	Deleted Scenes,Behind the Scen
2	ACE GOLDFINGER	A Astounding Epistle of a Database A...	2006	1	NULL	3	4.99	48	12.99	G	Trailers,Deleted Scenes
3	ADAPTATION HOLES	A Astounding Reflection of a Lumberj...	2006	1	NULL	7	2.99	50	18.99	NC-17	Trailers,Deleted Scenes
4	AFFAIR PREJUDICE	A Fanciful Documentary of a Frisbee ...	2006	1	NULL	5	2.99	117	26.99	G	Commentaries,Behind the Scene
5	AFRICAN EGG	A Fast-Paced Documentary of a Past...	2006	1	NULL	6	2.99	130	22.99	G	Deleted Scenes
6	AGENT TRUMAN	A Intrepid Panorama of a Robot And ...	2006	1	NULL	3	2.99	169	17.99	PG	Deleted Scenes
7	AIRPLANE SIERRA	A Touching Saga of a Hunter And a B...	2006	1	NULL	6	4.99	62	28.99	PG-13	Trailers,Deleted Scenes
8	AIRPORT POLLOCK	A Epic Tale of a Moose And a Girl who...	2006	1	NULL	6	4.99	54	15.99	R	Trailers
9	ALABAMA DEVIL	A Thoughtful Panorama of a Databas...	2006	1	NULL	3	2.99	114	21.99	PG-13	Trailers,Deleted Scenes
10	ALADDIN CALENDAR	A Action-Packed Tale of a Man And a ...	2006	1	NULL	6	4.99	63	24.99	NC-17	Trailers,Deleted Scenes
11	ALAMO VIDEOTAPE	A Boring Epistle of a Butler And a Cat...	2006	1	NULL	6	0.99	126	16.99	G	Commentaries,Behind the Scene
12	ALASKA PHANTOM	A Fanciful Saga of a Hunter And a Pas...	2006	1	NULL	6	0.99	136	22.99	PG	Commentaries,Deleted Scenes
13	ALI FOREVER	A Action-Packed Drama of a Dentist ...	2006	1	NULL	4	4.99	150	21.99	PG	Deleted Scenes,Behind the Scen
14	ALICE FANTASIA	A Emotional Drama of a A Shark And ...	2006	1	NULL	6	0.99	94	23.99	NC-17	Trailers,Deleted Scenes,Behind th
15	ALIEN CENTER	A Brilliant Drama of a Cat And a Mad ...	2006	1	NULL	5	2.99	46	10.99	NC-17	Trailers,Commentaries,Behind th
16	ALLEY EVOLUTION	A Fast-Paced Drama of a Robot And ...	2006	1	NULL	6	2.99	180	23.99	NC-17	Trailers,Commentaries

Designing Complex SQL Scripts

○ Lists all customers who have rented a film in the last 30 days.

```
3 • SELECT * FROM customer
4 JOIN RENTAL ON customer.customer_id=rental.rental_id
5 WHERE rental.rental_date >= NOW() - INTERVAL 30 DAY
6 ORDER BY rental.rental_date DESC;
7
```

customer_id	store_id	first_name	last_name	email	address_id	active	create_date	last_update	rental_id	rental_date	inventory_id	custo

○ Identifies the most rented film in the database.

```
7
8 • SELECT * FROM film
9 JOIN RENTAL ON film.film_id=rental.rental_id
10 ORDER BY rental_rate DESC;
11
12
13
```

film_id	title	description	release_year	language_id	original_language_id	rental_duration	renta
2	ACE GOLDFINGER	A Astounding Epistle of a Database A...	2006	1	NULL	3	4.99
7	AIRPLANE SIERRA	A Touching Saga of a Hunter And a B...	2006	1	NULL	6	4.99
8	AIRPORT POLLOCK	A Epic Tale of a Moose And a Girl who...	2006	1	NULL	6	4.99
10	ALADDIN CALENDAR	A Action-Packed Tale of a Man And a ...	2006	1	NULL	6	4.99
13	ALI FOREVER	A Action-Packed Drama of a Dentist ...	2006	1	NULL	4	4.99
20	AMELIE HELLFIGHTERS	A Boring Drama of a Woman And a S...	2006	1	NULL	4	4.99
21	AMERICAN CIRCUS	A Insightful Drama of a Girl And a Astr...	2006	1	NULL	3	4.99
28	ANTHEM LUKE	A Touching Panorama of a Waitress ...	2006	1	NULL	5	4.99
31	APACHE DIVINE	A Awe-Inspiring Reflection of a Pastrv...	2006	1	NULL	5	4.99

Leon_Ashley_Final_Report_Sakila :D

- Displays the total revenue generated per store.

```
14 -- JOINING 3 tables, we can get the total sells in each store
15 • SELECT store.store_id, SUM(payment.amount) AS total_sells FROM store
16 JOIN staff ON store.store_id = staff.store_id
17 JOIN payment ON staff.staff_id = payment.staff_id
18 GROUP BY store.store_id
19 ORDER BY total_sells DESC;
20
```

Result Grid			Filter Rows: <input type="text"/>	Export: 	Wrap Cell Content: 
	store_id	total_sells			
▶	2	33924.06			
	1	33482.50			

Understanding Transactions

```

2
3 • START TRANSACTION;
4
5 • INSERT INTO rental (rental_date, inventory_id, customer_id, return_date, staff_id)
6 VALUES (NOW(), 1000, 44, NOW(), 1);
7
8
9 • UPDATE inventory
10 SET last_update = NOW()
11 WHERE inventory_id = 1000;
12
13 • COMMIT;

```

[illegible]

Leon_Ashley_Final_Report_Sakila :D

```
9
10 -- Update the inventory
11 • UPDATE inventory
12   SET last_update = NOW()
13   WHERE inventory_id = 1000;
14
15 • COMMIT;
16
17 • select * from inventory;
```

Result Grid | Filter Rows: | Edit: | Export/Import:

	inventory_id	film_id	store_id	last_update
	997	222	2	2006-02-15 05:09:17
	998	222	2	2006-02-15 05:09:17
	999	223	2	2006-02-15 05:09:17
▶	1000	223	2	2025-03-16 18:31:01
	1001	224	1	2006-02-15 05:09:17
	1002	224	1	2006-02-15 05:09:17
	1003	225	1	2006-02-15 05:09:17
	1004	225	1	2006-02-15 05:09:17

Rolling Back Transactions

```
2
3 • START TRANSACTION;
4
5 -- We try to insert a new rental
6 • INSERT INTO rental (rental_date, inventory_id, customer_id, staff_id, return_date)
7   VALUES (NOW(), 4000, 99, 1, NULL);
8
9 -- rollback :sprakles:
10 • ROLLBACK;
11
12
13 -- I'm sorry i didnt understand the querie and ChatGPT didn't solve me the doubt (T...T) :(
```

320 19:29:32	START TRANSACTION	0 row(s) affected	0.000 sec
321 19:29:32	INSERT INTO rental (rental_date, inventory_id, customer...	1 row(s) affected	0.000 sec
322 19:29:32	ROLLBACK	0 row(s) affected	0.000 sec

The only challenge was understanding and the way I overcame that was by making my neurons work. (´ω´)?