Daycare Management System

Welcome!

This is an overview of the database and back-end structure behind my project For this project we've implemented a Daycare Management System, designed for efficient management of staff, parents, child attendance, learning activities, billing and more ...

1 Snippet of the ER Diagram

notifications
- id INT
- title VARCHAR(50)
- message TEXT
- recipient_type ENUM(...)
- recipient_id INT
- created_at TIMESTAMP
- Indexes

users
- id INT
- username VARCHAR(255)
- password VARCHAR(255)
- role ENUM(...)
- created_at TIMESTAMP
- email VARCHAR(255)
- phone VARCHAR(15)
- Indexes
- Triggers

active_teachers

child_billing_summary

attendance_today_summary

child_attendance_summary

billing
- id INT
- child_id INT
- amount DECIMAL(10,2)
- due_date DATE
- status ENUM(...)
- created_at TIMESTAMP
- Indexes

activity_resources
- id INT
- activity_id INT
- resource_type ENUM(...)
- resource TEXT
- created_at TIMESTAMP
- Indexes

staff
- id INT
- name VARCHAR(255)
- role VARCHAR(255)
- email VARCHAR(255)
- phone VARCHAR(15)
- created_at TIMESTAMP
- user_id INT
- Indexes
- Triggers

parents
- id INT
- name VARCHAR(100)
- email VARCHAR(100)
- phone VARCHAR(15)
- created_at TIMESTAMP
- Indexes
- Triggers

learning_activities
- id INT
- name VARCHAR(100)
- description TEXT
- activity_date DATE
- created_at TIMESTAMP
- Indexes

children
- id INT
- name VARCHAR(100)
- age INT
- parent_id INT
- phone VARCHAR(15)
- created_at TIMESTAMP
- Indexes
- Triggers

teacher_assignments
- id INT
- staff_id INT
- activity_id INT
- assigned_date DATE
- Indexes

attendance
- id INT
- date DATE
- child_id INT
- staff_id INT
- status ENUM(...)
- created_at TIMESTAMP
- Indexes
- Triggers

assessments
- id INT
- child_id INT
- activity_id INT
- score INT
- grade VARCHAR(10)
- feedback TEXT
- assessment_date DATE
- created_at TIMESTAMP
- Indexes

progress
- id INT
- child_id INT
- activity_id INT
- completion_percentage DECIMAL(5,2)
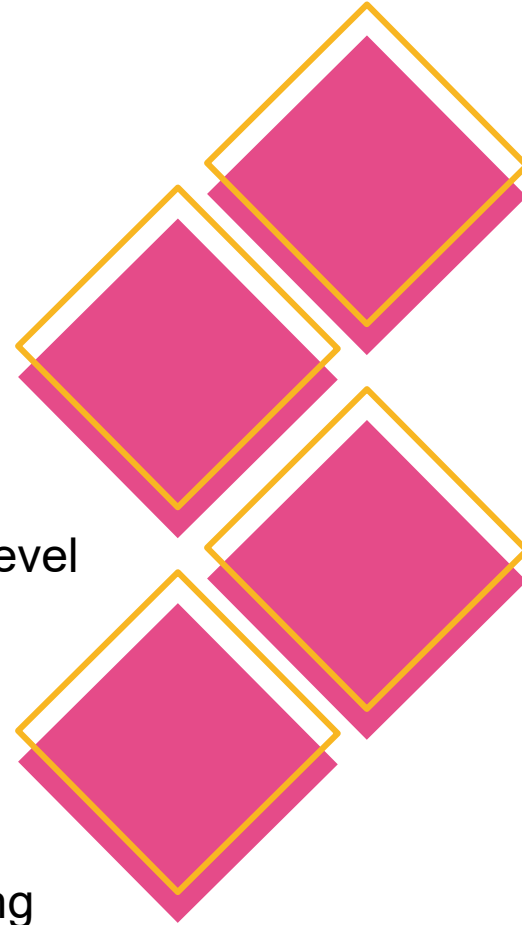- notes TEXT
- updated_at TIMESTAMP
- Indexes

3 **Isolation level**

This database uses MySQL and is using the default isolation level "REPEATABLE READ" to avoid dirty read and prevent non-repeatable reads while maintaining performance

Isolation levels are applied at the transaction level
Example of the store procedure

(create_billing_for_attendance)

To ensure that billing is created only once per attendance entry, even if multiple staff are using the system simultaneously.
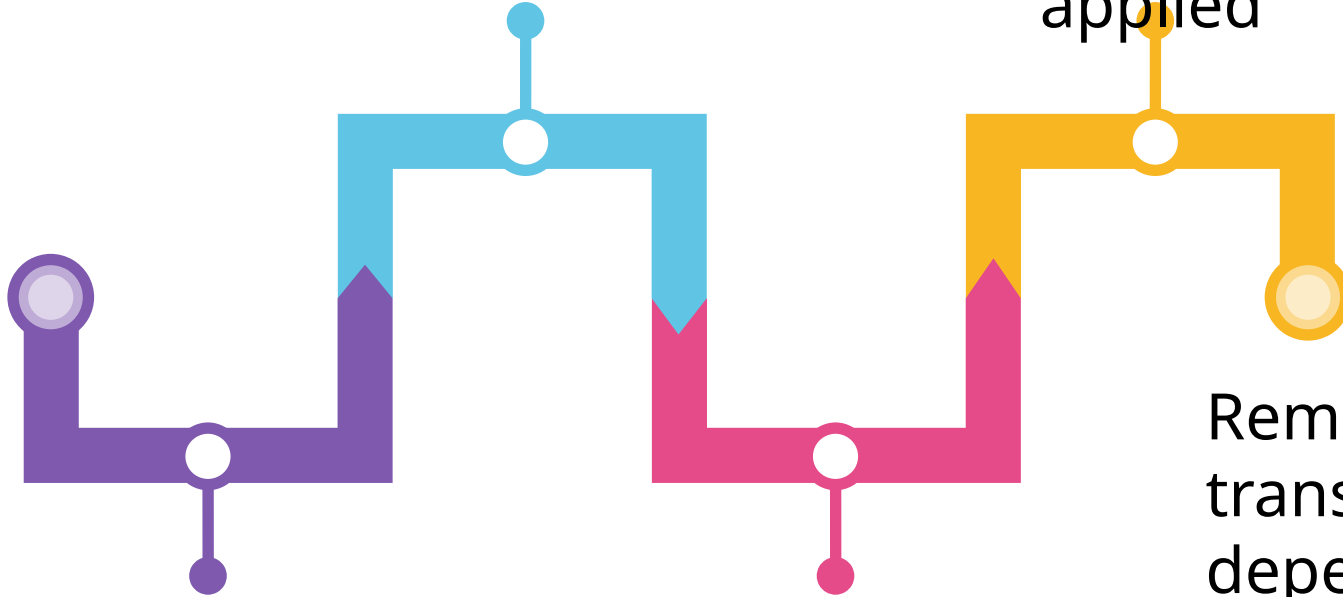
Primary keys

To ensure unique record

Every record in each table
is uniquely identified

Integrity
Enforcement

Foreign key

To maintain relational integrity
for proper connection between
tables

Logical relationships with foreign keys

Normalization

For Normalization the

3rd Normal Form is applied

No repeating data

All attributes depends on the primary key

Remove transitive dependencies

Database made of 12 tables each with the CRUD functionality

Users

- Contains the users separated by role.
- Admin and Staff

Parent

- Contains parents information

Children

- Contains child information

Attendance

- Keep track of children's attendance

2

View

These allow staff to retrieve meaningful data without complex queries.

create_billing_for_attendance
   - to get list of active staff.
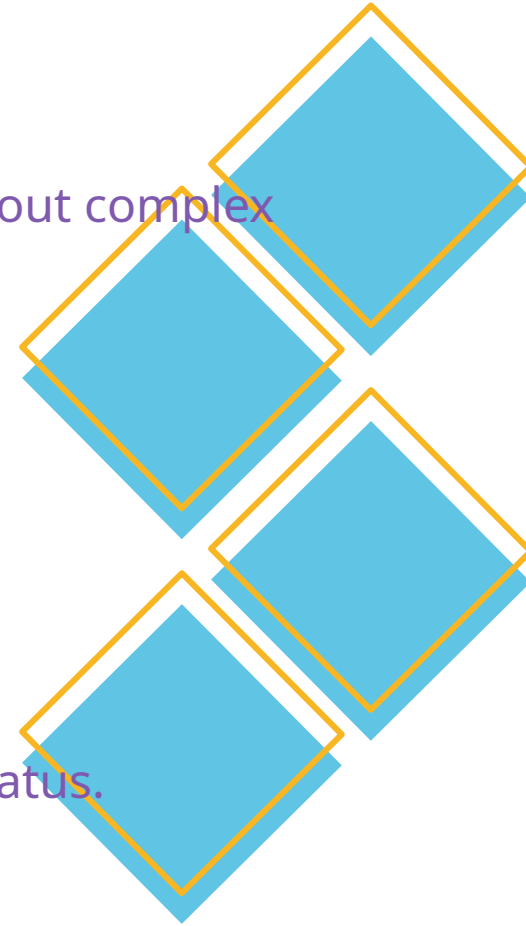create_billing_for_attendance
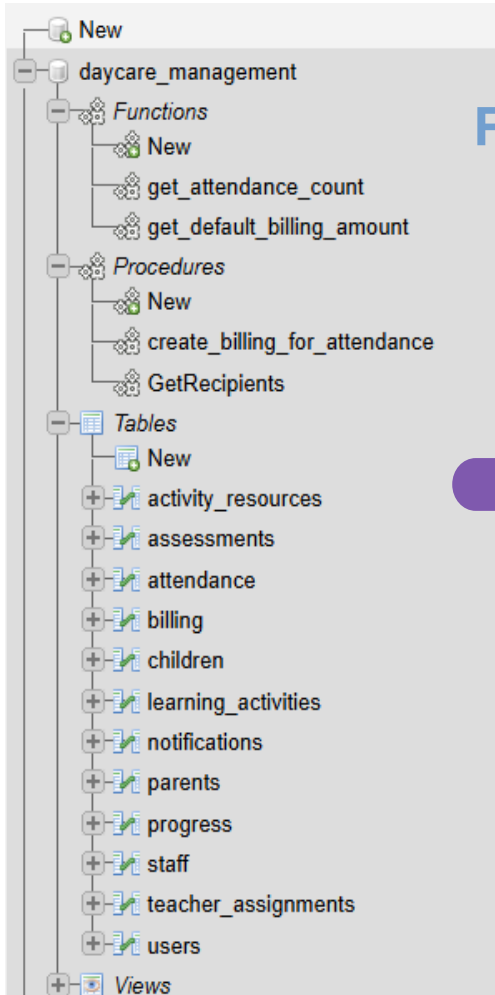   - for staff to view number of child present.
create_billing_for_attendance
   - for staff to view attendance summary
create_billing_for_attendance
   - Shows the total amount billed and payment status.

Function

get_attendance_count
to dynamically fetch
daily attendance count
for reporting

Procedure

create_billing_for_attendance

Procedure help automate logic
like automatically generate
billing when we take
attendance for child

Automatically insert data
to the staff table when the
user role is staff

and more

Triggers

Tree view (left panel):

- New
- daycare_management
  - Functions
    - New
    - get_attendance_count
    - get_default_billing_amount
  - Procedures
    - New
    - create_billing_for_attendance
    - GetRecipients
  - Tables
    - New
    - activity_resources
    - assessments
    - attendance
    - billing
    - children
    - learning_activities
    - notifications
    - parents
    - progress
    - staff
    - teacher_assignments
    - users
  - Views

Example Procedure Screenshot

create_billing_for_attendance

Details

Routine name	create_billing_for_attendance
Type	PROCEDURE

Parameters

	Direction	Name	Type	Length/Values	Options	
↕	IN	p_child	INT			⊖ Drop
↕	IN	p_date	DATE	---	---	⊖ Drop

Add parameter

Definition

```
1  BEGIN
2      DECLARE p_amount DECIMAL(10,2) DEFAULT 100.00;
3      DECLARE p_due_date DATE;
4
5      SET p_due_date = DATE_ADD(p_date, INTERVAL 7
   DAY);
6
7      INSERT INTO billing (child_id, amount, due_date,
   status, created_at)
8      VALUES (p_child_id, p_amount, p_due_date,
   'unpaid', NOW());
9  END
```

GetRecipients

Details

Routine name	GetRecipients
Type	PROCEDURE

Parameters

	Direction	Name	Type	Length/Values	Options	
↕	IN	type	VAF	50	Cha	⊖ Drop

Add parameter

Definition

```
1  BEGIN
2      IF type = 'Parent' THEN
3          SELECT id, name FROM parents;
4      ELSEIF type = 'Staff' THEN
5          SELECT id, name FROM staff;
6      ELSEIF type = 'Child' THEN
7          SELECT id, name FROM children;
8      END IF;
9  END
```

Thank you

Next - Live Application Demo