**INDEX SHEET**

[Basic ToDos : 2](#_gcsf07oz4a1n)

[**● Design the logical view using ER Diagrams with tools 2**](#_5irdxt1oyy9d)

[**● Design Enhanced ER diagram using Workbench 2**](#_itny3598um2k)

[**● Forward Engineer your EER diagrams n Workbench 2**](#_qnqjugg9mtq9)

[**● Design Test Cases/ SQL Queries to demonstrate the working (or) Implement a GUI 2**](#_9zx340m9pvu3)

[**(Python/Java/Web) to demonstrate the database. 2**](#_vhi8l3dzb9hs)

[**Blogging Platform: 2**](#_awstt1cjlexx)

[**a. Set up a database for blog posts, categories, tags, and user comments. 2**](#_aw66fmt23jaq)

[**b. Implement SQL queries to display popular blog posts, manage comments, and 2**](#_udawfs5vovcc)

[**categorize posts. 2**](#_iek2vdpn3esb)

**Aim:**

| **Basic ToDos :****● Design the logical view using ER Diagrams with tools****● Design Enhanced ER diagram using Workbench****● Forward Engineer your EER diagrams n Workbench****● Design Test Cases/ SQL Queries to demonstrate the working (or) Implement a GUI****(Python/Java/Web) to demonstrate the database.****Blogging Platform:****a. Set up a database for blog posts, categories, tags, and user comments.****b. Implement SQL queries to display popular blog posts, manage comments, and****categorize posts.** |
| --- |

**List of Entities & Attributes:**

| | **Entities** | **Attributes** | | --- | --- | | Event | Event\_ID, Event\_Name | | Post | Post\_ID, Post\_Article | | Tag | Tag\_ID, Tags | | Category | Category\_ID, Category\_Type | | Comment | Comment\_ID, Comment\_Data, User\_Name | | Organiser | Organiser\_ID, Organiser\_Name, Organiser\_Age, Organiser\_Experience | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |

**List of Relationships:**

| * Only one Organiser can organise only one Event. * Many Events can belongs to one Category. * One Event can have many Posts. * One or many Posts can have one or many Tags. * One Post can have many Comments. |
| --- |

**ER Diagram:**

|  |
| --- |

**EER Diagram:**

|  |
| --- |

**Schema:**

| -- MySQL Workbench Forward Engineering  SET @OLD\_UNIQUE\_CHECKS=@@UNIQUE\_CHECKS, UNIQUE\_CHECKS=0;  SET @OLD\_FOREIGN\_KEY\_CHECKS=@@FOREIGN\_KEY\_CHECKS, FOREIGN\_KEY\_CHECKS=0;  SET @OLD\_SQL\_MODE=@@SQL\_MODE, SQL\_MODE='ONLY\_FULL\_GROUP\_BY,STRICT\_TRANS\_TABLES,NO\_ZERO\_IN\_DATE,NO\_ZERO\_DATE,ERROR\_FOR\_DIVISION\_BY\_ZERO,NO\_ENGINE\_SUBSTITUTION';  -- -----------------------------------------------------  -- Schema mydb  -- -----------------------------------------------------  -- -----------------------------------------------------  -- Schema mydb  -- -----------------------------------------------------  CREATE SCHEMA IF NOT EXISTS `mydb` DEFAULT CHARACTER SET utf8 ;  USE `mydb` ;  -- -----------------------------------------------------  -- Table `mydb`.`Organiser`  -- -----------------------------------------------------  CREATE TABLE IF NOT EXISTS `mydb`.`Organiser` (  `Organiser\_ID` INT NOT NULL,  `Organiser\_Name` CHAR(100) NOT NULL,  `Organiser\_Age` INT NOT NULL,  `Organiser\_Experience` INT NOT NULL,  PRIMARY KEY (`Organiser\_ID`),  UNIQUE INDEX `Organiser\_ID\_UNIQUE` (`Organiser\_ID` ASC) VISIBLE)  ENGINE = InnoDB;  -- -----------------------------------------------------  -- Table `mydb`.`Category`  -- -----------------------------------------------------  CREATE TABLE IF NOT EXISTS `mydb`.`Category` (  `Category\_ID` INT NOT NULL,  `Category\_Type` CHAR(50) NOT NULL,  PRIMARY KEY (`Category\_ID`),  UNIQUE INDEX `Category\_ID\_UNIQUE` (`Category\_ID` ASC) VISIBLE)  ENGINE = InnoDB;  -- -----------------------------------------------------  -- Table `mydb`.`Event`  -- -----------------------------------------------------  CREATE TABLE IF NOT EXISTS `mydb`.`Event` (  `Event\_ID` INT NOT NULL,  `Event\_Name` VARCHAR(100) NOT NULL,  `Organiser\_Organiser\_ID` INT NOT NULL,  `Category\_Category\_ID` INT NOT NULL,  PRIMARY KEY (`Event\_ID`, `Organiser\_Organiser\_ID`, `Category\_Category\_ID`),  UNIQUE INDEX `Event\_ID\_UNIQUE` (`Event\_ID` ASC) VISIBLE,  INDEX `fk\_Event\_Organiser\_idx` (`Organiser\_Organiser\_ID` ASC) VISIBLE,  INDEX `fk\_Event\_Category1\_idx` (`Category\_Category\_ID` ASC) VISIBLE,  CONSTRAINT `fk\_Event\_Organiser`  FOREIGN KEY (`Organiser\_Organiser\_ID`)  REFERENCES `mydb`.`Organiser` (`Organiser\_ID`)  ON DELETE NO ACTION  ON UPDATE NO ACTION,  CONSTRAINT `fk\_Event\_Category1`  FOREIGN KEY (`Category\_Category\_ID`)  REFERENCES `mydb`.`Category` (`Category\_ID`)  ON DELETE NO ACTION  ON UPDATE NO ACTION)  ENGINE = InnoDB;  -- -----------------------------------------------------  -- Table `mydb`.`Post`  -- -----------------------------------------------------  CREATE TABLE IF NOT EXISTS `mydb`.`Post` (  `Post\_ID` INT NOT NULL,  `Post\_Article` VARCHAR(500) NOT NULL,  `Event\_Event\_ID` INT NOT NULL,  `Event\_Organiser\_Organiser\_ID` INT NOT NULL,  `Event\_Category\_Category\_ID` INT NOT NULL,  PRIMARY KEY (`Post\_ID`, `Event\_Event\_ID`, `Event\_Organiser\_Organiser\_ID`, `Event\_Category\_Category\_ID`),  UNIQUE INDEX `Post\_ID\_UNIQUE` (`Post\_ID` ASC) VISIBLE,  INDEX `fk\_Post\_Event1\_idx` (`Event\_Event\_ID` ASC, `Event\_Organiser\_Organiser\_ID` ASC, `Event\_Category\_Category\_ID` ASC) VISIBLE,  CONSTRAINT `fk\_Post\_Event1`  FOREIGN KEY (`Event\_Event\_ID` , `Event\_Organiser\_Organiser\_ID` , `Event\_Category\_Category\_ID`)  REFERENCES `mydb`.`Event` (`Event\_ID` , `Organiser\_Organiser\_ID` , `Category\_Category\_ID`)  ON DELETE NO ACTION  ON UPDATE NO ACTION)  ENGINE = InnoDB;  -- -----------------------------------------------------  -- Table `mydb`.`Tag`  -- -----------------------------------------------------  CREATE TABLE IF NOT EXISTS `mydb`.`Tag` (  `Tag\_ID` INT NOT NULL,  `Tags` VARCHAR(100) NOT NULL,  UNIQUE INDEX `Tag\_ID\_UNIQUE` (`Tag\_ID` ASC) VISIBLE,  PRIMARY KEY (`Tag\_ID`))  ENGINE = InnoDB;  -- -----------------------------------------------------  -- Table `mydb`.`Comment`  -- -----------------------------------------------------  CREATE TABLE IF NOT EXISTS `mydb`.`Comment` (  `Comment\_ID` INT NOT NULL,  `Comment\_Data` VARCHAR(500) NOT NULL,  `User\_Name` CHAR(100) NOT NULL,  `Post\_Post\_ID` INT NOT NULL,  `Post\_Event\_Event\_ID` INT NOT NULL,  `Post\_Event\_Organiser\_Organiser\_ID` INT NOT NULL,  `Post\_Event\_Category\_Category\_ID` INT NOT NULL,  PRIMARY KEY (`Comment\_ID`, `Post\_Post\_ID`, `Post\_Event\_Event\_ID`, `Post\_Event\_Organiser\_Organiser\_ID`, `Post\_Event\_Category\_Category\_ID`),  UNIQUE INDEX `Comment\_ID\_UNIQUE` (`Comment\_ID` ASC) VISIBLE,  INDEX `fk\_Comment\_Post1\_idx` (`Post\_Post\_ID` ASC, `Post\_Event\_Event\_ID` ASC, `Post\_Event\_Organiser\_Organiser\_ID` ASC, `Post\_Event\_Category\_Category\_ID` ASC) VISIBLE,  CONSTRAINT `fk\_Comment\_Post1`  FOREIGN KEY (`Post\_Post\_ID` , `Post\_Event\_Event\_ID` , `Post\_Event\_Organiser\_Organiser\_ID` , `Post\_Event\_Category\_Category\_ID`)  REFERENCES `mydb`.`Post` (`Post\_ID` , `Event\_Event\_ID` , `Event\_Organiser\_Organiser\_ID` , `Event\_Category\_Category\_ID`)  ON DELETE NO ACTION  ON UPDATE NO ACTION)  ENGINE = InnoDB;  -- -----------------------------------------------------  -- Table `mydb`.`Post\_has\_Tag`  -- -----------------------------------------------------  CREATE TABLE IF NOT EXISTS `mydb`.`Post\_has\_Tag` (  `Post\_Post\_ID` INT NOT NULL,  `Tag\_Tag\_ID` INT NOT NULL,  PRIMARY KEY (`Post\_Post\_ID`, `Tag\_Tag\_ID`),  INDEX `fk\_Post\_has\_Tag\_Tag1\_idx` (`Tag\_Tag\_ID` ASC) VISIBLE,  INDEX `fk\_Post\_has\_Tag\_Post1\_idx` (`Post\_Post\_ID` ASC) VISIBLE,  CONSTRAINT `fk\_Post\_has\_Tag\_Post1`  FOREIGN KEY (`Post\_Post\_ID`)  REFERENCES `mydb`.`Post` (`Post\_ID`)  ON DELETE NO ACTION  ON UPDATE NO ACTION,  CONSTRAINT `fk\_Post\_has\_Tag\_Tag1`  FOREIGN KEY (`Tag\_Tag\_ID`)  REFERENCES `mydb`.`Tag` (`Tag\_ID`)  ON DELETE NO ACTION  ON UPDATE NO ACTION)  ENGINE = InnoDB;  SET SQL\_MODE=@OLD\_SQL\_MODE;  SET FOREIGN\_KEY\_CHECKS=@OLD\_FOREIGN\_KEY\_CHECKS;  SET UNIQUE\_CHECKS=@OLD\_UNIQUE\_CHECKS; |
| --- |