

«ІНСТИТУТ ПРИКЛАДНОГО СИСТЕМНОГО АНАЛІЗУ» НАЦІОНАЛЬНОГО
ТЕХНІЧНОГО УНІВЕРСИТЕТУ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМ. ІГОРЯ СІКОРСЬКОГО»
КАФЕДРА МАТЕМАТИЧНИХ МЕТОДІВ СИСТЕМНОГО АНАЛІЗУ

КУРСОВА РОБОТА

з дисципліни «Організація баз даних»

на тему: «AIC театру»

Студента 3-го курсу групи КА-85
Спеціальність 124 Системний аналіз
Рачук Марії Русланівни
Керівник Афанасьєва І.В.
Національна оцінка _____
Кількість балів: ____ Оцінка: ECTS ____
Члени комісії _____

ЗМІСТ

ВСТУП	3
1. Постановка Задачі	5
2. Архітектура та інформаційне забезпечення БД	7
2.1 Аналіз функціонування та організаційні засади підприємства.....	8
2.2 Проектування структури бази даних... ..	8
2.3 Життєві цикли БД	8
3. Реалізація програмної взаємодії з БД.....	10
3.1 Інструкція користувача... ..	10
3.2 Реалізація механізмів SQL	10
3.2.1 SQL-запити... ..	10
3.2.2 Процедури і функції	10
3.3 Вимоги до апаратних і програмних засобів	13
3.4 Випробування розроблених програм	13
3.5 Опис тестової бази даних... ..	21
4. Висновок	26
5. Література	27
Додаток.....	28

Вступ

Актуальність.

Використання баз даних є однією з характерних рис більшості сучасних інформаційних систем. По своїй суті бази даних є тим, навколо чого і будується інформаційна система будь-якого підприємства. Тому теорії створення та практиці використання баз даних приділяється достатня увага протягом періоду функціонування інформаційних систем. Досить тривалий час основним типом були реляційні бази даних, які на сьогодні вже вважаються класичними. Проте розвиток інформаційних систем поставив перед сучасними базами даних завдання, вирішення яких неможливе в межах використання тільки реляційних баз даних. Крім класичних завдань, сучасні бази даних повинні забезпечувати багатомашинну обробку та зберігання великих обсягів інформації, оперативний аналіз даних, інтеграцію із мережею Інтернет, розмежування доступу користувачів до зберіганої інформації, захист інформації під час її передачі по мережі. Хоча на практиці і використовується чимало різноманітних баз даних, але для більшості з них існує велика кількість спільних ознак, як з погляду розробки, так і використання. Це дає можливість вивчати сучасні бази даних і відповідне прикладне та системне програмне забезпечення на прикладах, які, незважаючи на свою новизну, вже стали класичними. Як такі приклади вибрано загальні питання проектування, розробки та використання бази даних Microsoft SQL Server. Це пояснюється тим, що Microsoft SQL є однією із найпоширеніших і вдосконалених баз даних.

База даних - сукупність даних, що зберігаються у відповідності зі схемою даних, маніпулювання якими виконують відповідно до правил засобів моделювання даних.

СУБД - система управління базами даних - сукупність програм, призначених для управління БД і можливості отримання користувачами необхідної інформації з бази.

Мета.

Метою розробки є розробка бази даних театру, що відповідає всім поставленим вимогам.

Завдання.

Спроекувати базу даних та створити всі необхідні запити, для роботи з нею.

Практичне значення.

Відточення навиків SQL програмування.

Програмне забезпечення.

При виконанні роботи було використано таке програмне забезпечення: середовище розробки MySQL Workbench; операційна система Windows 10; база даних MySQL.

1. Постановка задачі

Постановка задачі полягає у проектуванні бази даних згідно текстового опису: Працівників театру можна підрозділити на акторів, музикантів, постановників і службовців. Кожна з перерахованих категорій має унікальні атрибути-характеристики і може підрозділятися (наприклад, постановники) на більш дрібні категорії. Театр очолює директор, в функції якого входять контроль за постановками вистав, затвердження репертуару, прийняття на роботу нових службовців, запрошення акторів і постановників. Актори, музиканти і постановники, які працюють в театрі, можуть виїжджати на гастролі. Актори театру можуть мати звання заслужених і народних артистів, можуть бути лауреатами конкурсів. Також акторами театру можуть бути і студенти театральних училищ. Кожен актор має свої вокальні і зовнішні дані (стать, вік, голос, зріст тощо), які можуть підходити для якихось ролей, а для якихось ні (не завжди жінка може зіграти чоловіка і навпаки).

Для постановки будь-якої вистави необхідно підібрати акторів на ролі і дублерів на кожен головну роль. Природно, що один і той же актор не може грати більше однієї ролі у виставі, але може грати кілька ролей у різних спектаклях. У спектакля також є режисер-постановник, художник-постановник, дирижер-постановник, автор. Вистави можна підрозділити за жанрами: музична комедія, трагедія, оперета тощо. З іншого боку, вистави можна поділити на дитячі, молодіжні тощо. У репертуарі театру вказується які вистави, в які дні і в який час будуть проходити, а також дати прем'єр. У касах театру можна заздалегідь придбати квитки або абонемент на будь-які вистави. Абонемент звичайно включає в себе квитки на вистави якогось конкретного автора, або конкретного жанру. Ціна квитка залежить від місця, і вистави. На прем'єру квитку вдвічі дорожче. Адміністрацією театру фіксується кількість проданих квитків на кожен спектакль.

Види запитів в інформаційній системі:

1. Отримати список і загальне число всіх працівників театру, акторів, музикантів, за стажем роботи в театрі, за статевою ознакою, роком народження, віком, ознакою наявності та кількістю дітей, розміром заробітної плати.
2. Отримати перелік і загальне число вистав, зазначених у репертуарі на даний сезон, вже зіграних вистав, спектаклів зазначеного жанру, коли-небудь зіграних у цьому театрі, за вказаний період.
3. Отримати перелік і загальне число всіх поставлених вистав, спектаклів зазначеного жанру, коли-небудь поставлених у цьому театрі, поставлених за вказаний період.
4. Отримати список авторів поставлених вистав, авторів, що жили в зазначеному столітті, авторів зазначеної країни, авторів вистав зазначеного жанру коли-небудь поставлених у цьому театрі, поставлених

за вказаний період часу.

5. Отримати перелік вистав зазначеного жанру, деякого автора, авторів позначеної країни, вистав, написаних у певному столітті, вперше поставлених на сцені зазначеного театру в означений період часу.
6. Отримати список акторів, що підходять за своїми даними на зазначену роль.
7. Отримати загальне число і список акторів театру, які мають звання, отримали їх за деякий період, на зазначених конкурсах, застатвою ознакою, за віком.
8. Отримати список акторів і постановників, які приїздили коли-небудь на гастролі до театру за вказаний період, перелік артистів, які їздили на гастролі за певний час з даними спектаклем.
9. Отримати список для зазначеної вистави: акторів, їх дублерів, імена режисера-постановника, художника-постановника, дирижера-постановника, авторів, дату прем'єри.
10. Отримати перелік і загальне число ролей, зіграних зазначеним актором за весь час, за деякий період часу, в спектаклях певного жанру, в спектаклях зазначеного режисера-постановника, у дитячих виставах.
11. Отримати відомості про кількість проданих квитків на всі вистави, на конкретний спектакль, на прем'єру, за вказаний період, у тому числі проданих попередньо.
12. Отримати загальну суму виручених грошей за вказаний спектакль, за деякий період часу.
13. Отримати перелік і загальне число вільних місць на всі вистави, на конкретний спектакль, на прем'єру.

2. Архітектура та інформаційне забезпечення БД

2.1 Аналіз функціонування і організаційні основи підприємства

Проаналізувавши запити, які необхідно буде реалізувати, можна зробити висновок, що база даних повинна включати наступні таблиці: Персонал, Постановка, Вистава, Автор, Актор, Постановник, Роль, Каса, Гастролю, Конкурс, ІншийПерсонал.

2.2 Проектування бази даних

Проектування бази даних можна відобразити за допомогою схеми, що зображена на рисунку 2.2:

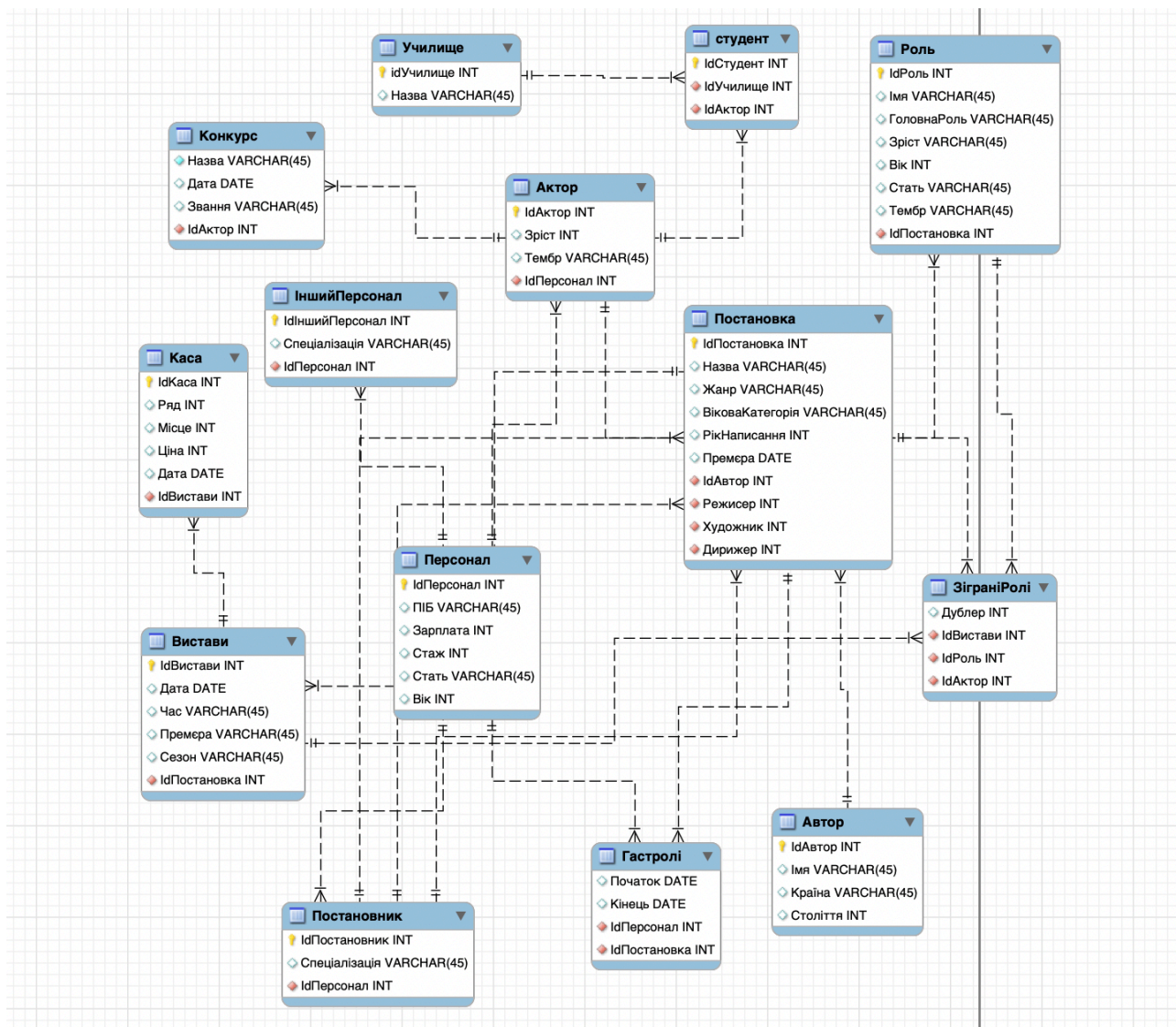


Рис 2.2. Схема проектування бази даних

2.3 Життєві цикли бази даних

1. Попереднє планування

На даному етапі була сформована модель структури бази даних, визначені сутності, проаналізовано зв'язки між сутностями на основі вимог до даної АІС.

2. Перевірка здійсненості

Для розробки даної системи необхідно мати персональний комп'ютер, на якому встановленні інструменти для розробки бази даних з використанням MySQL .
Наведені вимоги виконано, отже завдання є технічно здійсненим.

3. Визначення вимог

Вимоги до бази даних були визначені за допомоги завдання. База даних описує всі елементи, які там вказані.

4. Реалізація

Реалізація включає в себе написання програмного коду на SQL.

3. Реалізація програмної взаємодії з БД

3.1 Інструкція користувача

Для того, щоб користувач мав можливість редагувати та додавати дані в БД, йому потрібно встановити програмне забезпечення MySQL Workbench та MySQL Server.

3.2 Реалізація механізмів БД

3.2.1 SQL-запити

Для більш простого розуміння коду коду для користувачів та для мене самого я використовував процедури. Зазвичай, я просто робив SQL-запит я к процедуру. Усі процедури описані нижче

3.2.2 Процедури і функції

В ході реалізації АІС відповідно ТЗ були реалізовані наступні процедури:

Запит №1

personal() Повертає ім'я працівника та ідентифікаційний номер працівника

actors() Повертає ім'я актора та ідентифікаційний номер актора

musicians() Повертає список музикантів

experience () Повертає список працівників та їх стажу

sex(in s text) Повертає список працівників зазначеної статі

age(in n int) Повертає список працівників зазначеного віку

salary(in n double) Повертає список працівників із зазначеною зарплатою

Запит №2

performance() Повертає список вистав

played() Повертає список вже зіграних вистав

genre(in t text) Повертає список відіграних вистав зазначеного жанру

played_period(in b date, in e date) Повертає список відіграних вистав, зіграних за вказаний період

Запит №3

played_genre(in t text) Повертає список вже зіграних спектаклів певного жанру

played_period_genre(in b date, in e date, in t text) Повертає список спектаклів зазначеного жанру, зіграних за певний період.

Запит №4

author() Повертає список авторів

author_century(in n int) Повертає список авторів, що жили у вказаному столітті

country(in t text) Повертає список авторів , що жили в зазначеній країні

author_genre(in t text) Повертає список авторів , написаних в зазначеному жанрі

author_genre_played(in t text) Повертає список авторів , вистави яких коли-небудь були зіграні в театрі

author_genre_period(in t text, in b date, in e date) Повертає список авторів вистав, написаних в зазначеному жанрі за вказаний період

Запит №5

performance_genre(in t text) Повертає список вистав зазначеного жанру

performance_author(in a text) Повертає список вистав зазначеного автора

performance_country(in t text) Повертає список вистав авторів зазначеної країни

performance_century(in n text) Повертає список вистав написаних у зазначеному столітті

first_performance(in b date, in e date) Повертає список вистав вперше зіграних за вказаний період часу

Запит №6

role(in t text) Повертає список акторів, що підходять на зазначену роль

Запит №7

rank() Повертає список акторів зі званням

rank_getted(in t text) Повертає акторів що отримали звання за вказаний період

sex_rank(in t text) Повертає список акторів зазначеної статі та отриманим званням

rank_age(in n int) Повертає список акторів зазначеного віку та отриманим званням

Запит №8

tour_coming(in b date, in e date) Повертає список акторів та постановників , що приїжджали до театру за вказаний період

tour_outcoming(in t text) Повертає список акторів та постановників, що їздили на гастролі зі вказаним спектаклем

Запит №9

performance_list(in t text) Повертає список для зазначеної вистави: акторів, імена режисера-постановника, художника-постановника, дирижера-постановника, авторів, дату прем'єри.

Запит №10

role_actor(in t text) Повертає список ролей зіграних вказаним актором

role_period(in t text, in b date, in e date) Повертає список ролей зіграних вказаним актором за вказаний період

role_genre(in t text, in g text) Повертає список ролей зіграних вказаним артистом, в спектаклях зазначеного жанру

actor_regiser(in t text, in g text) Повертає список ролей зіграних вказаним артистом в постановках вказаного режисера

Запит №11

tickets() Повертає кількість проданих квитків на всі вистави

tickets_performance(in t text) Повертає кількість проданих квитків на вказану виставу

tickets_period(in b date, in e date) Повертає кількість проданих квитків за вказаний період

Запит №12

money_for_performance(in t text) Повертає загальну суму виручених грошей вказану виставу

money_for_period(in b date, in e date) Повертає загальну суму виручених грошей за вказаний період

Запит №13

place() Повертає перелік та загальне число вільних місць на всі вистави

place_performance(in t text) Повертає перелік та загальне число вільних місць на вказаний спектакль

3.3 Вимоги до програмних і апаратних засобів

Для роботи з базою даних необхідний комп'ютер або сервер, на якому встановлений MySQL Workbench версії 6.3 або вище.

3.4 Випробовування розроблених програм

Result Grid			Filter Rows: <input type="text"/>
	id	ПІБ	
▶	1	Петров А.О.	
	2	Иванова В.Є.	
	3	Сидоров П.Д.	
	4	Жук Ф.О.	
	5	Прищепа М.О.	
	6	Вишневська Я.Р.	
	7	Башкіров І.Ф.	
	8	Раскевич Ж.М.	
	9	Седих А.Є.	
	10	Порохницький Н.О.	
	11	Пасічний П.П.	
	12	Носов І.Р.	
	13	Пахарь І.О.	
	14	Гужевський М.О.	
	15	Калуж П.Р.	
	16	Авдієнко В.О	
	17	Клавдій Я.В.	
	Разом	17	

Рис 3.1 Результат на запит call personal();

	id	ПІБ	
▶	1	Петров А.О.	
	2	Иванова В.Є.	
	3	Жук Ф.О.	
	4	Пасічний П.П.	
	5	Носов І.Р.	
	6	Пахарь І.О.	
	7	Гужевський М.О.	
	8	Калуж П.Р.	
	11	Авдієнко В.О	
	15	Клавдій Я.В.	
	Разом	10	

Рис 3.2 Результат на запит call actors();

ПІБ	
► Сидоров П.Д.	
Разом:1	

Рис 3.3 Результат на запит call musicians();

ПІБ	Стаж	
► Петров А.О.	2	
Иванова В.Є.	5	
Сидоров П.Д.	1	
Жук Ф.О.	20	
Прищеп М.О.	7	
Вишнеvsька Я.Р.	1	
Башкіров І.Ф.	15	
Раскевич Ж.М.	3	
Седих А.Є.	3	
Порохницький Н.О.	6	
Пасічний П.П.	1	
Носов І.Р.	30	
Пахарь І.О.	5	
Гужевський М.О.	10	
Калуж П.Р.	2	
Авдієнко В.О	2	
Клавдій Я.В.	1	

Рис3.4 Результат на запит callexperience();

ПІБ	Стать	
► Петров А.О.	чол	
Сидоров П.Д.	чол	
Жук Ф.О.	чол	
Прищеп М.О.	чол	
Башкіров І.Ф.	чол	
Порохницький Н.О.	чол	
Пасічний П.П.	чол	
Носов І.Р.	чол	
Гужевський М.О.	чол	
Калуж П.Р.	чол	
Авдієнко В.О	чол	
Разом	11	

Рис3.5 Результат на запит call sex('чол');

	ПІБ	Вік	
►	Петров А.О.	25	
	Пахарь І.О.	25	
	Разом	2	

Рис3.6 Результат на запит call age(25);

	ПІБ	Зарплата	
►	Петров А.О.	10000	
	Сидоров П.Д.	10000	
	Вишневська Я.Р.	10000	
	Пасічний П.П.	10000	
	Носов І.Р.	10000	
	Авдієнко В.О	10000	
	Клавдій Я.В.	10000	
	Разом	7	

Рис3.7 Результат на запит call salary(10000);

	Назва	Дата	
►	Назва6	2020-10-10	
	Сто_Тисяч	2020-09-09	
	Разом:	2	

Рис3.8 Результат на запит call season('Осінь');

	Назва	Дата	
►	Корсиканка	2020-07-01	
	Сто_Тисяч	2020-06-02	
	Сто_Тисяч	2020-09-09	
	Мертві_Душі	2001-06-06	
	Мертві_Душі	2020-06-06	
	Назва4	2019-06-01	
	Назва5	2020-05-04	
	Назва6	2020-10-10	
	Разом:	8	

Рис3.9 Результат на запит call played();

Назва	
► Мертві_Душі	
Назва6	
Разом:3	

Рис3.10 Результат на запит call genre('Трагедія');

Назва	Дата	
► Корсиканка	2021-06-01	
Сто_Тисяч	2020-06-02	
Мертві_Душі	2021-06-03	
Мертві_Душі	2020-06-06	
Назва6	2020-10-10	
Сто_Тисяч	2020-09-09	
Корсиканка	2020-07-01	
Разом:	7	

Рис3.11 Результат на запит call played_period('2020.06.01', '2021.06.06');

Назва	Дата	
► Корсиканка	2020-07-01	
Сто_Тисяч	2020-06-02	
Сто_Тисяч	2020-09-09	
Назва4	2019-06-01	

Рис3.12 Результат на запит call played_genre('Комедія');

Назва	Дата	
► Корсиканка	2021-06-01	
Корсиканка	2020-07-01	
Сто_Тисяч	2020-06-02	
Сто_Тисяч	2020-09-09	
Назва4	2019-06-01	

Рис3.13 Результат на запит call played_period_genre('2000.05.08', '2021.06.01', 'Комедія');

Імя	
▶ Іржі_Губач	
Гоголь	
Карпенко-Карий	
Булгаков	
Жолі	
Бенавенте	

Рис3.14 Результат на запит call author();

Імя	
▶ Іржі_Губач	
Булгаков	

Рис3.15 Результат на запит call author_century(19);

Імя	
▶ Гоголь	
Карпенко-Карий	

Рис3.16 Результат на запит call country(' Україна');

Імя	
▶ Іржі_Губач	
Гоголь	
Булгаков	

Рис 3.17 Результат на запит call author_genre ('Комедія');

Імя	
▶ Гоголь	
Карпенко-Карий	
Булгаков	
Жолі	

Рис 3.18 Результат на запит call author_period('2000.05.07','2020.06.06');

Назва	
▶ Корсиканка	
Сто_Тисяч	
Назва4	

Рис 3.19 Результат на запит call performance_genre('Комедія');

Назва	
▶ Сто_Тисяч	

Рис 3.20 Результат на запит call performance_author('Гоголь');

Назва	
▶ Сто_Тисяч	
Мертві_Душі	

Рис 3.21 Результат на запит call performance_country('Україна');

Назва	
▶ Корсиканка	
Сто_Тисяч	
Назва4	
Назва6	

Рис 3.22 Результат на запит call performance_century(20);

Назва	Пр	Resets all so
▶ Сто_Тисяч	2019-04-02	
Мертві_Душі	2019-05-03	
Назва5	2020-06-03	
Назва6	2020-06-04	

Рис 3.23 Результат на запит call

first_performance('2019.01.01','2021.06.06');

	Імя	Вік	Стать	Тембр	Зріст
►	Бертран	25	чол	Баритон	Високий
	Петров А.О.	25	чол	Баритон	184

Рис 3.24 Результат на запит call role(' Бертран');

	ПІБ	Звання	
►	Петров А.О.	Заслужений_артист	
	Жук Ф.О.	Народний_співак	
	Носов І.Р.	Народний_артист	

Рис 3.25 Результат на запит call rank();

	ПІБ	Дата	
►	Петров А.О.	2020-05-05	
	Жук Ф.О.	2020-05-06	

Рис 3.26 Результат на запит call rank_period('2019.01.01', '2020.05.09');

	ПІБ	Звання	
►	Петров А.О.	Заслужений_артист	
	Носов І.Р.	Народний_артист	

Рис 3.27 Результат на запит call rank_getted('Кращий_актор');

	ПІБ	Назва	Звання
►	Петров А.О.	Кращий_актор	Заслужений_артист
	Жук Ф.О.	Кращий_бас	Народний_співак
	Носов І.Р.	Кращий_актор	Народний_артист

Рис 3.28 Результат на запит call rank_sex('чол');

	ПІБ	Звання	
►	Петров А.О.	Заслужений_артист	

Рис 3.29 Результат на запит call rank_age(25);

	ПІБ	Актор	
►	Петров А.О.	Актор	
	Пасічний П.П.	Актор	
	Вишневська Я.Р.	Постановник	

Рис 3.30 Результат на запит call tour_coming('2020-05-01', '2020-05-020');

	ПІБ	Початок	Кінець	
►	Калуж П.Р.	2020-04-01	2020-04-12	
	Носов І.Р.	2020-04-01	2020-04-12	

Рис 3.31 Результат на запит call tour_outcoming('2020-04-01', '2020-05-020', 'Мертві_Душі');

	Дата Прем'єри	Премєра	
►	Дата Прем'єри	2015-03-01	
	Режисер	Порохницький Н.О.	
	Художник	Вишневська Я.Р.	
	Дирижер	Седих А.Є.	
	Жозефіна	Иванова В.Є.	
	Герасим	Жук Ф.О.	

Рис 3.32 Результат на запит call performance_list('Корсиканка');

	Роль	
►	Жозефіна	

Рис 3.33 Результат на запит call role_actor('Иванова В.Є.');

Роль	
▶ Жозефіна	

Рис 3.34 Результат на запит `call role_period('Иванова В.Є.', '2020.06.01', '2020.06.02');`

Роль	
▶ Жозефіна	

Рис 3.35 Результат на запит `all role_genre('Иванова В.Є.', 'Комедія');`

Роль	
▶ Жозефіна	

Рис 3.36 Результат на запит `call actor_regiser('Иванова В.Є.', 'Порохницький Н.О.');`

Назва	К-ть проданих квитків	Дата
Сто_Тисяч	6	2020-06-02
Сто_Тисяч	1	2020-09-09
Мертві_Душі	4	2021-06-03
Мертві_Душі	1	2001-06-06
Мертві_Душі	1	2020-06-06
Назва4	1	2019-06-01
Назва5	1	2020-05-04
Назва6	1	2020-10-10

Рис 3.43 Результат на запит `call tickets();`

	Назва	К-ть проданих квитків	Дата
►	Корсиканка	5	2021-06-01
	Корсиканка	1	2020-07-01

Рис 3.37 Результат на запит call
tickets_performance('Корсиканка');

	Назва	К-ть проданих квитків	Дата вистави
►	Сто_Тисяч	4	2020-06-02
	Мертві_Душі	4	2021-06-03
	Корсиканка	1	2021-06-01

Рис 3.38 Результат на запит call tickets_period('2020-06-02',
'2020-06-03');

	Назва	К-ть проданих квитків	Виручені кошти	Дата
►	Корсиканка	5	1010	2021-06-01
	Корсиканка	1	200	2020-07-01
	Усього:	6	1210	

Рис 3.39 Результат на запит call
money_for_performance('Корсиканка');

	Назва	Виручені кошти	Дата
►	Сто_Тисяч	700	2020-06-02
	Мертві_Душі	840	2021-06-03
	Корсиканка	100	2021-06-01
	Усього:	1640	

Рис 3.40 Результат на запит call
money_for_period('2020-06-02', '2020-06-03');

	Назва	Кількість вільних місць	Дата	
►	Корсиканка	5	2021-06-01	
	Корсиканка	9	2020-07-01	
	Сто_Тисяч	4	2020-06-02	
	Сто_Тисяч	9	2020-09-09	
	Мертві_Душі	6	2021-06-03	
	Мертві_Душі	9	2001-06-06	
	Мертві_Душі	9	2020-06-06	
	Назва4	9	2019-06-01	
	Назва5	9	2020-05-04	
	Назва6	9	2020-10-10	
	Усього:	88		

Рис 3.41 Результат на запит call place();

	Назва	Кількість вільних місць	Дата	
►	Корсиканка	5	2021-06-01	
	Корсиканка	9	2020-07-01	

Рис 3.42 Результат на запит call place_performance('Корсиканка');

3.5 Опис тестової бази даних Тестові таблиці:

	IdАвтор	Ім'я	Країна	Століття	
	1	Іржі_Губач	Чехія	19	
►	2	Гоголь	Україна	20	
	3	Карпенко-Карий	Україна	20	
	4	Булгаков	Росія	19	
	5	Жолі	Франція	20	
	6	Бенавенте	Іспанія	20	
	NULL	NULL	NULL	NULL	

Рис 3.43 Таблиця «Автор»

IdАктор	Зріст	Тембр	IdПерсонал	
3	177	Бас	4	
4	166	Тенор	11	
5	199	Бас	12	
6	155	Контральто	13	
7	175	Баритон	14	
8	178	Баритон	15	
11	189	Бас	16	
15	167	Сопрано	17	
NULL	NULL	NULL	NULL	

Рис 3.44 Таблиця «Актор»

Початок	Кінець	IdПерсонал	IdПостановка	
2020-05-05	2020-05-12	11	1	
2020-04-01	2020-04-12	12	3	
2020-04-01	2020-04-12	15	3	
2020-05-05	2020-05-12	6	1	
2020-05-15	2020-05-22	7	2	
2020-04-01	2020-04-12	8	3	

Рис 3.45 Таблиця «Гастролі»

Дублер	IdВистави	IdРоль	IdАктор	
0	1	3	3	
0	4	4	6	
0	2	5	5	
0	3	6	1	
1	3	5	5	

Рис 3.46 Таблиця «ЗіграніРолі»

IdІншийПерсонал	Спеціалізація	IdПерсонал	
1	Музикант	3	
2	Адміністратор	5	
NULL	NULL	NULL	

Рис 3.47 Таблиця «ІншийПерсонал»

	IdКаса	Ряд	Місце	Ціна	Дата	IdВистави
►	1	3	1	400	2020-06-01	1
	2	4	2	110	2020-06-01	1
	3	1	6	200	2020-06-02	2
	4	2	4	200	2020-06-03	3
	5	6	1	150	2020-06-01	1
	6	6	1	200	2020-06-02	2
	7	4	2	200	2020-06-02	2
	8	3	5	300	2020-06-03	3
	9	3	1	250	2020-06-01	1
	10	1	5	100	2020-06-02	2
	11	2	4	170	2020-06-03	3
	12	6	5	170	2020-06-03	3
	13	4	3	100	2020-06-03	1
	14	4	8	100	2020-09-09	2
	15	2	7	100	2020-09-09	2
	16	3	1	200	2001-06-06	4
	17	3	1	200	2020-06-06	5
	18	1	1	100	2020-10-10	6
	19	1	1	200	2019-06-01	7
	20	1	1	200	2020-07-01	8
	21	1	1	200	2020-09-09	9
	22	1	1	200	2020-06-04	10

Рис 3.48 Таблиця «Каса»

	Назва	Дата	Звання	IdАктор
►	Кращий_актор	2020-05-05	Заслужений_артист	1
	Кращий_бас	2020-05-06	Народний_співак	3
	Кращий_актор	2020-05-10	Народний_артист	5

Рис 3.49 Таблиця «Конкурс»

	IdПерсонал	ПІБ	Зарплата	Стаж	Стать	Вік
►	1	Петров А.О.	10000	2	чол	25
	2	Иванова В.Є.	12000	5	жін	33
	3	Сидоров П.Д.	10000	1	чол	22
	4	Жук Ф.О.	18000	20	чол	52
	5	Прищепа М.О.	15000	7	чол	37
	6	Вишневська Я.Р.	10000	1	жін	20
	7	Башкіров І.Ф.	25000	15	чол	47
	8	Раскевич Ж.М.	16200	3	жін	26
	9	Седих А.Є.	15000	3	жін	26
	10	Порохницький Н.О.	18000	6	чол	29
	11	Пасічний П.П.	10000	1	чол	19
	12	Носов І.Р.	10000	30	чол	62
	13	Пахарь І.О.	17500	5	жін	25
	14	Гужевський М.О.	20000	10	чол	40
	15	Калуж П.Р.	13000	2	чол	23
	16	Авдієнко В.О	10000	2	чол	20
	17	Клавдій Я.В.	10000	1	жін	20

Рис 3.50 Таблиця «Персонал»

IdПостановка	Назва	Жанр	ВіковаКатегорія	РікНаписання	Премєра	IdАвтор	Режисер	Художник	Дирижер
3	Мертві Душі	Трагедія	12+	2001	2019-05-03	3	4	2	3
4	Назва4	Комедія	6+	1988	2005-06-08	4	5	2	1
5	Назва5	Драма	12+	1788	2020-06-03	5	5	2	3
6	Назва6	Трагедія	18+	1967	2020-06-04	6	5	2	3

Рис 3.51 Таблиця «Постановка»

IdПостановник	Спеціалізація	IdПерсонал
3	Дирижер	7
4	Режисер	8
5	Режисер	10

Рис 3.52 Таблиця «Постановник»

IdВистави	Дата	Час	Премєра	Сезон	IdПостановка
1	2021-06-01	18:00	1	Літо	1
2	2020-06-02	19:00	0	Літо	2
3	2021-06-03	20:00	0	Літо	3
4	2001-06-06	13:00	0	Літо	3
5	2020-06-06	12:00	0	Літо	3
6	2019-06-01	20:00	1	Літо	4
7	2020-05-04	13:00	0	Літо	5
8	2020-10-10	11:00	0	Осінь	6
9	2020-09-09	12:00	0	Осінь	2
10	2020-07-01	13:00	0	Літо	1
11	2021-07-01	13:00	1	Літо	2

Рис 3.53 Таблиця «Вистави»

IdРоль	Ім'я	ГоловнаРоль	Зріст	Вік	Стать	Тембр	IdПостановка
3	Герасим	1	Середній	35	чол	Бас	2
4	Параска	1	Низький	35	жін	Сопрано	2
5	Мотря	0	Середній	25	жін	Сопрано	2
6	Плюшкін	0	Середній	40	чол	Бас	3

Рис 3.54 Таблиця «Роль»

	IdСтудент	IdУчилище	IdАктор
▶	1	1	11
	2	2	15
	NULL	NULL	NULL

Рис 3.55 Таблиця «Студент»

	idУчилище	Назва
▶	1	Коледж театру і кіно
	2	Університет театру
	NULL	NULL

Рис 3.56 Таблиця «Училище»

4. Висновки

Задача з проектування бази даних для театру – виконана. Виконані всі вимоги, із запитами, які повинна виконати система. Були покращені навички програмування, а також був отриманий досвід роботи з БД.

5. Література

1. Гарсиа-Молина Г., Ульман Д., Уидом Д. Системы баз данных. Полный курс.: Пер. с англ. - М.: Издательский дом "Вильямс", 2004. - 1088 с.
2. Дейт К. Дж. Введение в системы баз данных. :Пер. с англ. – 6-е изд. – К.: Диалектика, 1998. – 784 с.
3. Калянов Г.Н. CASE-технологии. Консалтинг в автоматизации бизнес- процессов. – 3-е изд. – М.: Горячая линия-Телеком, 2002. - 320 с.
4. Карпова Т.С. Базы данных: модели, разработка, реализация. – СПб.: Питер, 2001. – 304 с.
5. Когаловский М.Р. Энциклопедия технологий баз данных. – М.: Финансы и статистика, 2002. – 800 с.
6. Конноли Т., Бегг К., Страчан А. Базы данных: проектирование, реализация и сопровождение. Теория и практика., 2-е изд.: Пер. с англ. – М.: Издательский дом «Вильямс», 2001. – 1120 с.

5. Додаток Лістинг запитів

```
use Treatre_db;

delimiter $$

drop procedure if exists personal$$

create procedure personal()

begin

select idПерсонал as 'id' , ПІБ from Персонал

union select 'Разом', count(idПерсонал) from Персонал;

end$$

#call personal();


drop procedure if exists actors$$

create procedure actors()

begin

select Актор.idАктор as 'id', Персонал.ПІБ from Актор, Персонал

where Актор.idПерсонал = Персонал.idПерсонал

union select 'Разом', count(idАктор) from Актор;

end$$

#call actors();


drop procedure if exists musicians$$

create procedure musicians()

begin

declare c int default 0;

select count(idІншийПерсонал) into c from ІншийПерсонал where

ІншийПерсонал.Спеціалізація='Музикант';

select Персонал.ПІБ from Персонал, ІншийПерсонал where
```

Персонал.idПерсонал=ІншийПерсонал.idПерсонал and
ІншийПерсонал.Спеціалізація='Музикант'

union select concat('Разом:', convert(c, char));

end\$\$

#call musicians();

drop procedure if exists experience\$\$

create procedure experience()

begin

select ПІБ , Стаж from Персонал;

end\$\$

#call experience();

drop procedure if exists sex\$\$

create procedure sex(in s text)

begin

declare c int default 0;

select count(idПерсонал) into c from Персонал

where Персонал.Стать=s; select ПІБ , Стать from Персонал

where Персонал.Стать=s

union select 'Разом', c;

end\$\$

#call sex('чол');

drop procedure if exists age\$\$

create procedure age(in n int)

begin

declare c int default 0;

```

select count(idПерсонал) into c from Персонал where Персонал.Вік=n ;

select ПІБ, Вік from Персонал where Персонал.Вік=n

union select 'Разом', c;

end$$

#call age(25);


drop procedure if exists salary$$

create procedure salary(in n int)

begin

declare c int default 0;

select count(idПерсонал) into c from Персонал where Персонал.Зарплата=n ;

select ПІБ, Зарплата from Персонал where Персонал.Зарплата=n

union select 'Разом', c;

end$$

#call salary(10000);

# Task1 finished -----

use Treatre_db;

delimiter $$

drop procedure if exists performance$$

create procedure performance() begin

declare c int default 0;

select count(idПостановка) into c from Постановка;

select Назва from Постановка

union select concat('Разом:', convert(c, char));

end$$

#call performance();

```



```

drop procedure if exists season$$

create procedure season(in t text) begin

declare c int default 0;

select count(idВистави) into c from Вистави where Вистави.Сезон=t and (to_days(now())-
to_days(Вистави.Дата)) >=0;

select Постановка.Назва , Вистави.Дата from Постановка, Вистави

where Вистави.Сезон=t and Постановка.idПостановка=Вистави.idПостановка and
to_days(now())- to_days(Вистави.Дата) >=0

union select 'Разом:', c;

end$$

#call season('Осінь');

```

```

drop procedure if exists played$$

create procedure played()

begin

declare c int default 0;

select count(idВистави) into c from Вистави where to_days(now())-to_days(Вистави.Дата) >=0;

select Постановка.Назва , Вистави.Дата from Постановка, Вистави

where Постановка.idПостановка=Вистави.idПостановка and to_days(now())-
to_days(Вистави.Дата) >=0

union select 'Разом:', c;

end$$

#call played();

```

```

drop procedure if exists genre$$

create procedure genre(in t text)

begin

declare c int default 0;

select count(Постановка.idПостановка) into c from Постановка, Вистави

```

```
where Постановка.idПостановка=Вистави.idПостановка and Постановка.Жанр=t and  
to_days(now())-to_days(Вистави.Дата) >=0;
```

```
select Назва from Постановка, Вистави
```

```
where Постановка.idПостановка=Вистави.idПостановка and Постановка.Жанр=t and  
to_days(now())-to_days(Вистави.Дата) >=0
```

```
union select concat('Разом:', convert(c,char)); end$$
```

```
call genre('Трагедія');
```

```
drop procedure if exists played_period$$
```

```
create procedure played_period(in b date, in e date)
```

```
begin
```

```
declare c int default 0;
```

```
select count(idВистави) into c from Вистави where Дата between b and e;
```

```
select Постановка.Назва, Вистави.Дата from Постановка, Вистави
```

```
where Вистави.Дата between b and e and Постановка.idПостановка=Вистави.idПостановка
```

```
union select 'Разом:', c;
```

```
end$$
```

```
call played_period('2020.06.01', '2021.06.06');
```

```
#Task2-----
```

```
use Treatre_db;
```

```
delimiter $$
```

```
drop procedure if exists played$$
```

```
create procedure played() begin
```

```
declare c int default 0;
```

```
select count(idВистави) into c from Вистави where to_days(now())-to_days(Дата) >=0;
```

```
select Постановка.Назва , Вистави.Дата from Постановка, Вистави
```

```
where Постановка.idПостановка=Вистави.idПостановка and to_days(now())- to_days(Дата) >=0
```

```
union select 'Разом:', c; end$$
```

```
call played();
```

```
drop procedure if exists played_genre$$
```

```
create procedure played_genre(in t text)
```

```
begin
```

```
select Постановка.Назва , Вистави.Дата from Постановка, Вистави where
```

```
Постановка.idПостановка=Вистави.idПостановка and Постановка.Жанр=t
```

```
and to_days(now())-to_days(Вистави.Дата) >=0; end$$
```

```
call played_genre('Комедія');
```

```
drop procedure if exists played_period_genre$$
```

```
create procedure played_period_genre(in b date, in e date, in t text )
```

```
begin
```

```
select Постановка.Назва, Вистави.Дата from Постановка, Вистави
```

```
where Вистави.Дата between b and e and Постановка.idПостановка=Вистави.idПостановка and  
Постановка.Жанр=t;
```

```
end$$
```

```
call played_period_genre('2000.05.08', '2021.06.01', 'Комедія');
```

```
#Task3-----
```

```
use Treatre_db;
```

```
delimiter $$
```

```
drop procedure if exists author$$
```

```
create procedure author()
```

```
begin
```

```
select Імя from Автор;
```

```
end$$
```

```
#call author();
```

```
drop procedure if exists author_century$$
```

```
create procedure author_century(in n int)
```

```
begin
```

```
select Імя from Автор where Століття = n;
```

```
end$$
```

```
#call author_century(19);
```

```
drop procedure if exists country$$
```

```
create procedure country(in t text)
```

```
begin
```

```
select Імя from Автор where Країна=t;
```

```
end$$
```

```
#call country('Україна');
```

```
drop procedure if exists author_genre$$
```

```
create procedure author_genre(in t text)
```

```
begin
```

```
select DISTINCT Автор.Імя from Автор, Постановка
```

```
where Постановка.idАвтор=Автор.idАвтор and Постановка.Жанр = t;
```

```
end$$
```

```
#call author_genre('Комедія');
```

```
drop procedure if exists author_genre_played$$
```

```
create procedure author_genre_played(in t text)
```

```
begin
```

```
select DISTINCT Автор.Імя from Автор, Постановка, Вистави
```

```
where Постановка.idАвтор=Автор.idАвтор and Постановка.Жанр = t and
Постановка.idПостановка=Вистави.idПостановка
```

```
and to_days(now())- to_days(Вистави.Дата) >=0;
```

```
end$$
```

```
#call author_genre_played('Комедія');
```

```
drop procedure if exists author_period$$
```

```
create procedure author_period( in b date, in e date) begin
```

```
select DISTINCT Автор.Імя from Автор, Постановка, Вистави
```

```
where Постановка.idАвтор=Автор.idАвтор and
Постановка.idПостановка=Вистави.idПостановка and Вистави.Дата between b and e;
```

```
end$$
```

```
#call author_period('2000.05.07','2020.06.06');
```

```
#Task4-----
```

```
use Treatre_db;
```

```
delimiter $$
```

```
drop procedure if exists performance_genre$$
```

```
create procedure performance_genre(in t text)
```

```
begin
```

```
select Назва from Постановка
```

```
where Постановка.Жанр=t;
```

```
end$$
```

```
#call performance_genre('Комедія');
```

```
drop procedure if exists performance_author$$
```

```
create procedure performance_author(in a text)
```

```
begin
```

```
select Постановка.Назва from Постановка, Автор
```

```
where Постановка.idАвтор = Автор.idАвтор and Автор.Імя=a;
```

```
end$$
```

```
#call performance_author('Гоголь');
```

```
drop procedure if exists performance_country$$
```

```
create procedure performance_country(in t text)
```

```
begin
```

```
select Постановка.Назва from Постановка, Автор
```

```
where Постановка.idАвтор=Автор.idАвтор and Автор.Країна=t;
```

```
end$$
```

```
#call performance_country('Україна');
```

```
drop procedure if exists performance_century$$
```

```
create procedure performance_century(in n int)
```

```
begin
```

```
select Назва from Постановка
```

```
where Постановка.РікНаписання between 100*(n-1) and 100*n;
```

```
end$$
```

```
#call performance_century(20);
```

```
drop procedure if exists first_performance$$
```

```
create procedure first_performance(in b date, in e date)
```

```
begin
```

```
select Назва, Премєра from Постановка
```

```
where Постановка.Премєра between b and e;
```

```
end$$
```

```
#call first_performance('2019.01.01','2021.06.06');
```

#Task5-----

use Treatre_db;

delimiter \$\$

drop procedure if exists role\$\$ create procedure role(in t text) begin

declare c, min,max int default 0;

declare r, age, age1 int default 0;

declare sex, tembr , height text default '';

select idРоль into r from Роль where Роль.Імя=t;

select Стать into sex from Роль where Роль.Імя=t;

select Вік into age from Роль where Роль.Імя=t;

select Зріст into height from Роль where Роль.Імя=t;

select if(age<20, 20, age) into age1;

select

case

when height='Високий' then 184

when height='Середній' then 174

when height='Низький' then 164

end

into max;

select Тембр into tembr from Роль where Роль.Імя=t;

select t as Імя, age as 'Вік',sex as 'Стать', tembr as 'Тембр', height as 'Зріст' union

select Персонал.ПІБ, Персонал.Вік, Персонал.Стать, Актор.Тембр,

convert(Актор.Зріст, char) from Персонал, Актор

where Персонал.idПерсонал=Актор.idПерсонал and Персонал.Стать=sex and
Персонал.Вік=age1

and Актор.Тембр=tembr and Актор.Зріст between (max-10) and max;

end\$\$

```

#call role('Бертран');

#Task6-----

use Treatre_db;

delimiter $$

drop procedure if exists rankk$$

create procedure rankk()

begin

select Персонал.ПИБ, Конкурс.Звання from Персонал, Актор, Конкурс

where Конкурс.idАктор=Актор.idАктор and Актор.idПерсонал=Персонал.idПерсонал;

end$$

#call rankk();


drop procedure if exists rank_period$$

create procedure rank_period(in b date, in e date)

begin

select Персонал.ПИБ, Конкурс.Дата from Персонал, Актор, Конкурс where

Конкурс.idАктор=Актор.idАктор and Актор.idПерсонал=Персонал.idПерсонал

and Конкурс.Дата between b and e;

end$$

#call rank_period('2019.01.01', '2020.05.09');


drop procedure if exists rank_getted$$

create procedure rank_getted(in t text)

begin

select Персонал.ПИБ, Конкурс.Звання from Персонал, Актор, Конкурс

where Конкурс.idАктор=Актор.idАктор and Актор.idПерсонал=Персонал.idПерсонал and

Назва=t;

end$$

```



```
#call rank_getted('Кращий_актор');
```

```
drop procedure if exists sex_rank$$
```

```
create procedure sex_rank(in t text)
```

```
begin
```

```
select Персонал.ПІБ, Конкурс.Назва, Конкурс.Звання from Персонал, Актор, Конкурс
```

```
where Конкурс.idАктор=Актор.idАктор and Актор.idПерсонал=Персонал.idПерсонал and  
Персонал.Стать=t;
```

```
end$$
```

```
#call sex_rank('чол');
```

```
drop procedure if exists rank_age$$
```

```
create procedure rank_age(in n int)
```

```
begin
```

```
select Персонал.ПІБ, Конкурс.Звання from Персонал,
```

```
Актор, Конкурс where Конкурс.idАктор=Актор.idАктор
```

```
and Актор.idПерсонал=Персонал.idПерсонал and Персонал.Вік=n;
```

```
end$$
```

```
#call rank_age(25);
```

```
#Task7-----
```

```
use Treatre_db;
```

```
delimiter $$
```

```
drop procedure if exists tour_coming$$
```

```
create procedure tour_coming(in b date, in e date)
```

```
begin
```

```
select Персонал.ПІБ, "Актор" from Персонал, Актор , Гастролі
```

```
where Гастролі.idПерсонал=Персонал.idПерсонал and  
Персонал.idПерсонал=Актор.idПерсонал
```

and Початок between b and e and

Кінець between b and e union

```
select Персонал.ПІБ, "Постановник" from Персонал, Постановник, Гастролі
```

```
where Гастролі.idПерсонал=Персонал.idПерсонал and  
Персонал.idПерсонал=Постановник.idПерсонал
```

and Початок between b and e

and Кінець between b and e;

```
end$$
```

```
call tour_coming('2020-05-01', '2020-05-020');
```

```
drop procedure if exists tour_outcoming$$
```

```
create procedure tour_outcoming(in b date, in e date, in t text)
```

```
begin
```

```
declare l int default 0;
```

```
select idПостановка into l from Постанова
```

```
where Постанова.Назва=t; select distinct(Персонал.ПІБ), Початок, Кінець
```

```
from Персонал, Актор, Гастролі, Постанова
```

```
where Гастролі.idПерсонал=Персонал.idПерсонал and  
Персонал.idПерсонал=Актор.idПерсонал
```

and Гастролі.idПостановка=l and Початок between b and e

and Кінець between b and e;

```
end$$
```

```
call tour_outcoming('2020-04-01', '2020-05-020', 'Мертві_Душі');
```

```
#Task8-----
```

```
use Treatre_db;
```

```
delimiter $$
```

```
drop procedure if exists performance_list$$
```

```
create procedure performance_list(in t text)
```

```
begin
```

```

declare n , p int default 0;

select idПостановка into n from Постановка

where Постановка.Назва=t; select idВистави into p from Вистави

where idПостановка= n limit 1; select "Дата Прем'єри", Премєра from Постановка

where Постановка.Назва=t union

select "Режисер", Персонал.ПІБ from Постановник, Персонал, Постановка

where Постановник.idПерсонал=Персонал.idПерсонал and
Постановка.Режисер=Постановник.idПостановник

and Постановка.idПостановка=n union select "Художник", Персонал.ПІБ

from Постановник, Персонал, Постановка where
Постановник.idПерсонал=Персонал.idПерсонал

and Постановка.Художник=Постановник.idПостановник and Постановка.idПостановка=n
union

select "Дирижер", Персонал.ПІБ from Постановник, Персонал, Постановка

where Постановник.idПерсонал=Персонал.idПерсонал and

Постановка.Дирижер=Постановник.idПостановник and Постановка.idПостановка=n union

select Роль.Імя, Персонал.ПІБ from Роль, ЗіграніРолі, Актор, Персонал

where ЗіграніРолі.idВистави=p and ЗіграніРолі.Дублер=0 and ЗіграніРолі.idРоль=Роль.idРоль

and ЗіграніРолі.idАктор=Актор.idАктор and Актор.idПерсонал=Персонал.idПерсонал;

end$$

call performance_list('Корсиканка');

#Task9-----

use Treatre_db;

delimiter $$

drop procedure if exists role_actor$$

create procedure role_actor(in t text)

begin

declare r, k int default 0;

```

```
select idПерсонал into r from Персонал where ПІБ=t;

select idАктор into k from Актор where Актор.idПерсонал=r;

select Роль.Імя as 'Роль' from ЗіграніРолі, Роль
where ЗіграніРолі.idАктор=k and ЗіграніРолі.idРоль=Роль.idРоль;

end$$

#call role_actor('Иванова В.Є.');
```

```
drop procedure if exists role_period$$

create procedure role_period(in t text, in b date, in e date)

begin

declare r, k int default 0;

select idПерсонал into r from Персонал where ПІБ=t;

select idАктор into k from Актор where Актор.idПерсонал=r;

select Роль.Імя as 'Роль' from ЗіграніРолі, Роль , Вистави where

ЗіграніРолі.idАктор=k and ЗіграніРолі.idРоль=Роль.idРоль

and ЗіграніРолі.idВистави=Вистави.idВистави and Вистави.Дата between b and e;

end$$

#call role_period('Иванова В.Є.', '2020.06.01', '2020.06.02'); #+1day
```

```
drop procedure if exists role_genre$$

create procedure role_genre(in t text, in g text)

begin

declare r, k int default 0;

select idПерсонал into r from Персонал where ПІБ=t;

select idАктор into k from Актор where Актор.idПерсонал=r;

select Роль.Імя as 'Роль' from ЗіграніРолі, Роль , Вистави, Постановка

where ЗіграніРолі.idАктор=k and ЗіграніРолі.idРоль=Роль.idРоль and

ЗіграніРолі.idВистави=Вистави.idВистави and
```

```

Вистави.idПостановка=Постановка.idПостановка and Постанова.Жанр= g;

end$$

#call role_genre('Иванова В.Є.', 'Комедія');#Трагедія


drop procedure if exists actor_regiser$$

create procedure actor_regiser(in t text, in g text)

begin

declare k int default 0;

select idАктор into k from Актор, Персонал

where Персонал.ПІБ=t and Актор.idПерсонал=Персонал.idПерсонал;

select distinct(Роль.Імя) as 'Роль' from ЗіграніРолі, Роль , Вистави, Постанова, Постановник,
Персонал

where Персонал.ПІБ=g and Постановник.idПерсонал=Персонал.idПерсонал

and Постанова.Режисер=Постановник.idПостановник and ЗіграніРолі.idАктор=k

and ЗіграніРолі.idВистави=Вистави.idВистави and
Вистави.idПостановка=Постановка.idПостановка

and ЗіграніРолі.idРоль=Роль.idРоль;

end$$

call actor_regiser('Иванова В.Є.', 'Порохницький Н.О.');
```

#Task10-----

```

use Treatre_db;

delimiter $$

drop procedure if exists tickets$$

create procedure tickets()

begin

select Постанова.Назва, count(Каса.idВистави)as 'К-ть проданих квитків',

Вистави.Дата from Каса, Вистави, Постанова

where Вистави.idВистави=Каса.idВистави and
Вистави.idПостановка=Постановка.idПостановка

```

drop procedure if exists money_for_performance\$\$

```

create procedure money_for_performance(in t text)

begin

select Постановка.Назва as 'Назва', count(Каса.idВистави) as 'К-ть проданих квитків',
sum(Каса.Ціна) as 'Виручені кошти', Вистави.Дата

from Постановка, Вистави, Каса where Постановка.Назва =t

and Постановка.idПостановка=Вистави.idПостановка and Вистави.idВистави=Каса.idВистави

group by Каса.idВистави union

select 'Усього:', count(Каса.idВистави) as 'К-ть проданих квитків', sum(Каса.Ціна) , " from
Постановка, Вистави, Каса

where Постановка.Назва = t and Постановка.idПостановка=Вистави.idПостановка and
Вистави.idВистави=Каса.idВистави;

end$$

#call money_for_performance('Корсиканка');

drop procedure if exists money_for_period$$

create procedure money_for_period(in b date, in e date)

begin

select Постановка.Назва, sum(Каса.Ціна) as 'Виручені кошти' , Вистави.Дата from Каса,
Вистави, Постанова

where Каса.Дата between b

and e and Постанова.idПостановка=Вистави.idПостановка and
Вистави.idВистави=Каса.idВистави

group by Каса.idВистави union

select 'Усього:', sum(Каса.Ціна) as 'Виручені кошти', ' ' from Каса

where Каса.Дата between b and e;

end$$

#call money_for_period('2020-06-02', '2020-06-03');# do 2020-08-03

#Task12 — — — — —

use Treatre_db;

delimiter $$

```

```

drop procedure if exists place$$

create procedure place()

begin

declare t int default 0;

select idВистави into t from Вистави order by idВистави desc limit 1;

select Постановка.Назва as 'Назва', (10 - count(Каса.idВистави)) as 'Кількість вільних місць',
Вистави.Дата as 'Дата'

from Каса,Вистави, Постановка

where Постановка.idПостановка=Вистави.idПостановка and
Вистави.idВистави=Каса.idВистави

group by Каса.idВистави union

select 'Усього:' , (t*10-count(idКаса)), " from Каса;

end$$

#call place();

```

```

drop procedure if exists place_performance$$

create procedure place_performance(in t text)

begin

select t as 'Назва', (10 - count(Каса.idВистави)) as 'Кількість вільних місць',
Вистави.Дата from Каса,Вистави, Постановка

where Постановка.Назва=t

and Постановка.idПостановка=Вистави.idПостановка and Вистави.idВистави=Каса.idВистави

group by Каса.idВистави;

end$$

#call place_performance('Корсиканка');

```

```

#Task13-----

```