POLITECHNIKA RZESZOWSKA
im. IGNACEGO ŁUKASIEWICZA

Katedra
Informatyki i Automatyki
Politechnika Rzeszowska

Bazy danych

Sprawozdanie nr 4

pt.: „System zarządzania projektami”

Data wykonania: 28-11-2020r

Adrian Pikul
158843 3EF-ZI

Spis treści

Spis treści 2
1. Cel pracy 3
2. Przebieg Pracy 3
1.1. Etap 1 3
1.2. Etap 2 3
1.3. Etap 3 3
3. Wnioski 20
4. Spis rysunków i tabel 20

1. Cel pracy

Projekt, który chciałbym zrealizować, będzie bazą danych pomagającej firmie zarządzać projektami oraz pracownikami. Będzie on pozawalał na takie funkcjonalności jak administracje użytkownikami, klientami oraz projektami tj. dodanie, usunięcie oraz modyfikacja. Dodatkową opcją będą karty pracy.

2. Przebieg Pracy

1.1. Etap 1

Określenie funkcji bazy danych i ich priorytetu

W dzisiejszych czasach dane gromadzone przez różne firmy, czy instytucje są na wagę złota i muszą być gdzieś gromadzone. Kluczowe jest filtrowanie tych danych w jak najszybszym czasie. W moim projekcie baza danych będzie przechowywała dane klientów, użytkowników oraz, projektów.
Projekt uwzględnia 3 rodzaje użytkowników:

- Administrator - Pełen dostęp do całej bazy danych
- Księgowa – Prawa pracownika + dodatkowo pełna modyfikacja tabeli ProjFinance, Contracts, Workers.
- Pracownik – Posiada prawa do modyfikacji oraz dodawania klientów, oraz projektów. Uzupełnia kartę pracy.

Wybór technologii i typu bazy danych

Projekt zostanie wykonany przy wykorzystaniu relacyjnych baz danych. Zdecydowałem się korzystać z MySQL ponieważ jest to darmowe oprogramowanie na licencji open source i doskonale sprawdza się przy małych projektach.

Wybór narzędzi

Swoją pracę oprę głównie na narzędziu MySQL Workbench, ponieważ jest to graficzne narzędzie do projektowania i administrowania baz danych.
Według mnie doskonale się sprawdzi podczas tego projektu.

Repozytorium GitHub

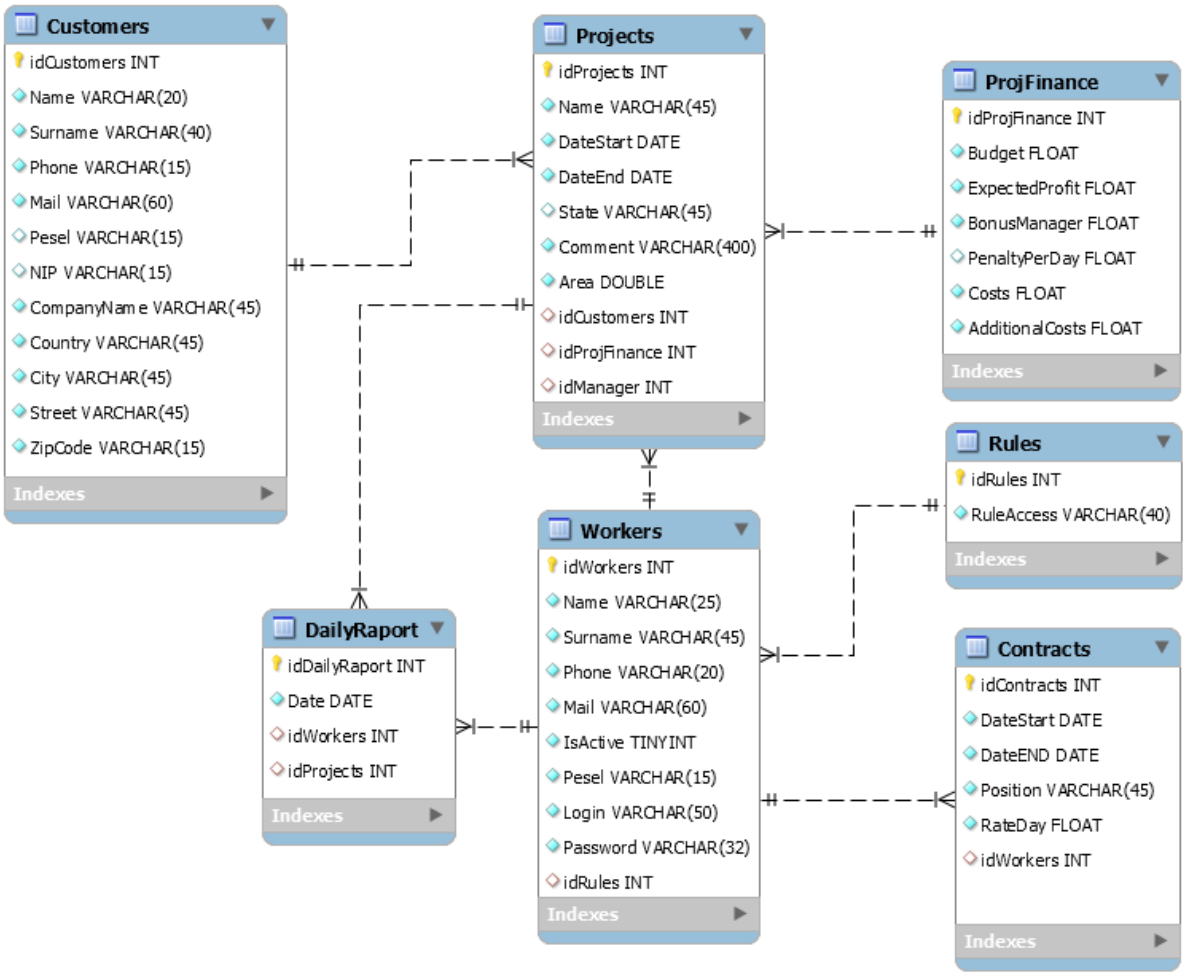
Link do repozytorium na GitHub

https://github.com/adrianpikul/projectDB

1.2. Etap 2

Prezentacja diagramu ERD

Diagram stworzyłem wykorzystując program MySQL Workbench.



Rysunek 1. Baza danych - Diagram ERD

Opis tabel bazy danych i ich funkcji

Customers – Przechowuje szczegółowe dane klientów. Jest w relacji jeden do wielu z tabelą Projekty, ponieważ, jeden klient może być w posiadaniu kilku projektów

Projects – Przechowuje dane odnośnie projektów. Jest w relacji jeden do wiele do jednego, ponieważ jeden projekt może mieć tylko jednego klienta.
Jest też w drugiej relacji jeden do wielu z tabelą ProjFinanse, ponieważ jeden projekt może mieć jedne finanse, a ProjFinanse mogą mieć kilka projektów.
ProjFinance – Przechowuje informacje odnośnie zysków i kosztów firmy. W koszty wliczamy premie dla kierownika projektu, dodatkowe koszty oraz koszty standardowe. Pod uwagę jest też wzięta ewentualna kara za opóźnienia projektu.

DailyRaport – Zadaniem tej tabeli jest przechowywanie informacji, które później wykorzystamy do policzenia kosztów projektu względem ilości pracujących osób i ilości poświęconych dni. Coś na zasadzie karty pracy.

Workers – Tabela przechowująca informacje o pracownikach, ważnymi rekordami są tutaj pracLogin i pracPassword -> Mają one być pomocniczne przy logowaniu bezpośrednio do aplikacji, nie do bazy danych. Jest to powiązane z tabelą Role. Podczas logowanie aplikacja miałaby pobierać login i hasło oraz role i według tego dostosowywać funkcjonalność. W tym projekcie aplikacja dla klientów jest nieuwzględniona.

Rules – Jest odpowiedzialna za przydzielanie funkcjonalności użytkownikom, rozróżniam następujące role:
- Administrator (szef)
- Pracownik
- Księgowa

Contracts – Tabela zawierająca podstawowe informacje na temat długości trwania umowy, data rozpoczęcia, stanowiska oraz wynagrodzenia.

Opis tabel bazy danych i ich funkcji

Skrypt tworzący bazę danych dostępny pod adresem:
https://github.com/adrianpikul/projectDB/blob/main/projectDB.sql

Fragment skryptu. Wycinek kodu tworzącegy tabelę ProjFinance.

```
DROP TABLE IF EXISTS `mydb`.`ProjFinance` ;


CREATE TABLE IF NOT EXISTS `mydb`.`ProjFinance` (
  `idProjFinance` INT NOT NULL AUTO_INCREMENT,
  `Budget` FLOAT NOT NULL,
  `ExpectedProfit` FLOAT NOT NULL,
  `BonusManager` FLOAT NOT NULL,
  `PenaltyPerDay` FLOAT NULL,
  `Costs` FLOAT NOT NULL,
  `AdditionalCosts` FLOAT NOT NULL,
  PRIMARY KEY (`idProjFinance`),
  UNIQUE INDEX `idProjFinance_UNIQUE` (`idProjFinance` ASC)
VISIBLE)
ENGINE = InnoDB;
```

Wycinek 1. Kod tworzący tabelę ProjFinance

Prezentacja problemów w realizacji

Jednym z problemów było wymyślenie sposobu na karty pracy, aby zliczać koszty danego projektu. Finalnie założenie było takie, że dzienny raport pracy ma być uzupełniany codziennie, i na podstawie daty z raportu odpytam rekordy z okresu trwania projektu, sprawdzając w tym czasie pensje pracownika w tabeli Contracts.

1.3. Etap 3

Własny skrypt generujące dane do bazy danych

W tym etapie stworzyłem własny skrypt, który generował rekordy do bazy. Dzięki temu moja baza posiadała pokaźną ilość rekordów, aby umożliwić mi testowanie zapytań. Całość skryptów tworzących rekordy dla danej tabeli dostępna w linku poniżej nazwy tabeli.

INSERT Contracts – 1909 rekordów

https://github.com/adrianpikul/projectDB/blob/main/INSERT%20Contracts%20-%201909_records.sql

INSERT Customers – 1200 rekordów

https://github.com/adrianpikul/projectDB/blob/main/INSERT%20Customers%20-%201200_records.sql

INSERT DailyRaport – 15 917 rekordów

https://github.com/adrianpikul/projectDB/blob/main/INSERT%20DailyRaport-%2015917_records.sql

INSERT ProjFinance – 179 rekordów

https://github.com/adrianpikul/projectDB/blob/main/INSERT%20ProjFinance%20-%20179_records.sql

INSERT Projects – 179 rekordów

https://github.com/adrianpikul/projectDB/blob/main/INSERT%20Projects-%20179_records.sql

INSERT Rules – 3 rekordy

https://github.com/adrianpikul/projectDB/blob/main/INSERT%20Rules%20-%203_records.sql

INSERT Workers – 90 rekordów

https://github.com/adrianpikul/projectDB/blob/main/INSERT%20Workers%20-%2090_records.sql

```
INSERT INTO Rules(RuleAccess)
VALUES ('PRACOWNIK'), ('KSIEGOWA'), ('SZEF');
```

Wycinek 2. Przykładowy kod do umieszczania danych w tabeli.

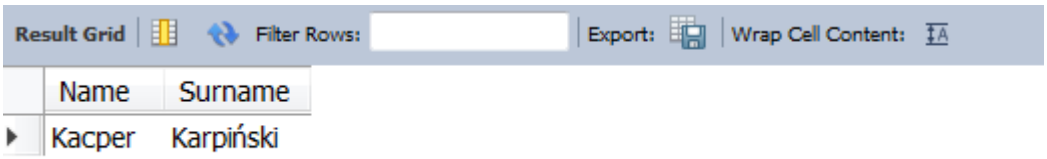
Przykładowe zapytania SELECT w celu sprawdzenia poprawności danych.

Wszystkie przykładowe zapytania dostępne pod adresem:

https://github.com/adrianpikul/projectDB/blob/main/SELECTS.sql

```
SELECT Name, Surname FROM projDB.Workers WHERE idWorkers = 2;
```

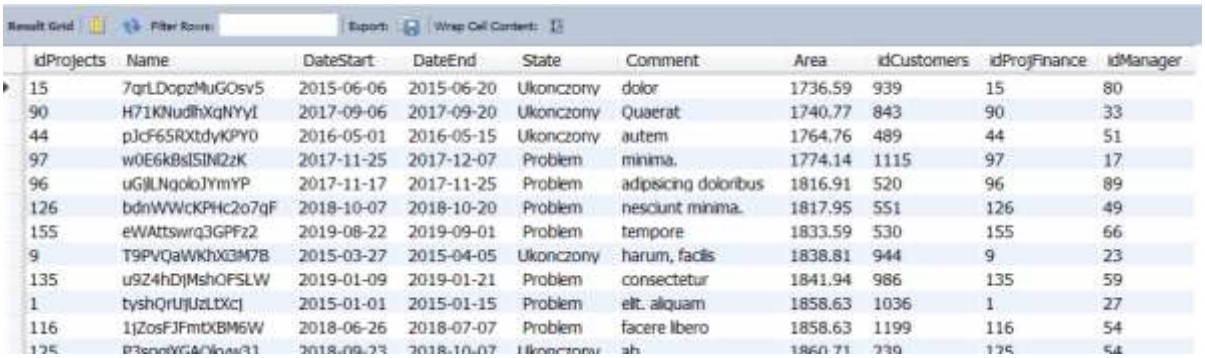
Wycinek 3 Pobierz imię i nazwisko pracownika.



Rysunek 2 Rysunek przedstawia rezultat wykonania powyższego polecenia SELECT.

```
SELECT * FROM Projects WHERE Area > 1700 ORDER BY Area;
```

Wycinek 4 Pobierz wszystkie projekty, gdzie powierzchnia jest większa niż 1700.

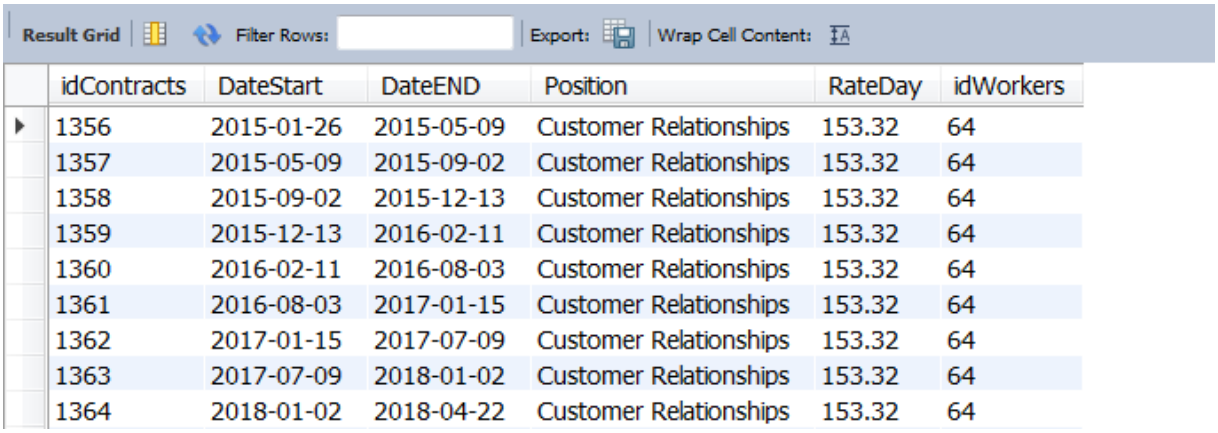


idProjects	Name	DateStart	DateEnd	State	Comment	Area	idCustomers	idProjFinance	idManager
15	7qrLDopzMuGOsv5	2015-06-06	2015-06-20	Ukonczony	dolor	1736.59	939	15	80
90	H71KNudlhXqNYyI	2017-09-06	2017-09-20	Ukonczony	Quaerat	1740.77	843	90	33
44	pJcF65RXtdyKPY0	2016-05-01	2016-05-15	Ukonczony	autem	1764.76	489	44	51
97	w0E6kBslSIN2zK	2017-11-25	2017-12-07	Problem	minima.	1774.14	1115	97	17
96	uGjlLNqoloJYmYP	2017-11-17	2017-11-25	Problem	adipisicing doloribus	1816.91	520	96	89
126	bdnWWcKPHc2o7qF	2018-10-07	2018-10-20	Problem	nesciunt minima.	1817.95	551	126	49
155	eWAttswrq3GPFz2	2019-08-22	2019-09-01	Problem	tempore	1833.59	530	155	66
9	T9PVQaWKhXi3M7B	2015-03-27	2015-04-05	Ukonczony	harum, facilis	1838.81	944	9	23
135	u9Z4hDjMshOFSLW	2019-01-09	2019-01-21	Problem	consectetur	1841.94	986	135	59
1	tyshQrUjUzLtXcj	2015-01-01	2015-01-15	Problem	elit. aliquam	1858.63	1036	1	27
116	1jZosFJFmtXBM6W	2018-06-26	2018-07-07	Problem	facere libero	1858.63	1199	116	54

Rysunek 3 Rysunek przedstawia fragment rezultatu wykonania powyższego polecenia SELECT.

```
SELECT * FROM Contracts WHERE idWorkers = 64 ORDER BY
DATESTART;
```

Wycinek 5 Pobierz wszystkie umowy dla pracownika, sortuj według daty.



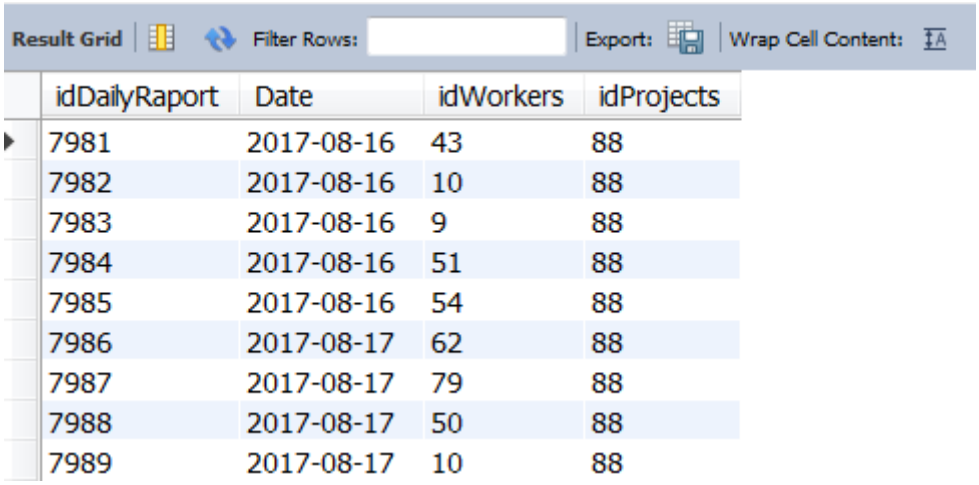
Result Grid | Filter Rows: | Export: | Wrap Cell Content:

idContracts	DateStart	DateEND	Position	RateDay	idWorkers
1356	2015-01-26	2015-05-09	Customer Relationships	153.32	64
1357	2015-05-09	2015-09-02	Customer Relationships	153.32	64
1358	2015-09-02	2015-12-13	Customer Relationships	153.32	64
1359	2015-12-13	2016-02-11	Customer Relationships	153.32	64
1360	2016-02-11	2016-08-03	Customer Relationships	153.32	64
1361	2016-08-03	2017-01-15	Customer Relationships	153.32	64
1362	2017-01-15	2017-07-09	Customer Relationships	153.32	64
1363	2017-07-09	2018-01-02	Customer Relationships	153.32	64
1364	2018-01-02	2018-04-22	Customer Relationships	153.32	64

Rysunek 4 Rysunek przedstawia fragment rezultatu wykonania powyższego polecenia SELECT.

```
SELECT * FROM DailyRaport WHERE  idProjects = 88 ORDER BY
idDailyRaport ;
```

Wycinek 6 Pobierz wszystkie dzienne raporty dla projektu, sortuj według id raportu.



Result Grid | Filter Rows: | Export: | Wrap Cell Content:

idDailyRaport	Date	idWorkers	idProjects
7981	2017-08-16	43	88
7982	2017-08-16	10	88
7983	2017-08-16	9	88
7984	2017-08-16	51	88
7985	2017-08-16	54	88
7986	2017-08-17	62	88
7987	2017-08-17	79	88
7988	2017-08-17	50	88
7989	2017-08-17	10	88

Rysunek 5 Rysunek przedstawia fragment rezultatu wykonania powyższego polecenia SELECT.

```
SELECT * FROM mydb.DailyRaport WHERE Date BETWEEN
STR_TO_DATE('2,1,2019','%d,%m,%Y') AND
STR_TO_DATE('18,1,2020','%d,%m,%Y');
```

Wycinek 7 Pobierz wszystkie dzienne raporty pomiędzy datami.

idDailyRaport	Date	idWorkers	idProjects
11843	2019-01-02	42	134
11844	2019-01-02	30	134
11845	2019-01-02	47	134
11846	2019-01-02	55	134
11847	2019-01-03	62	134
11848	2019-01-03	21	134
11849	2019-01-03	11	134
11850	2019-01-03	20	134
11851	2019-01-04	36	134
11852	2019-01-04	49	134
11853	2019-01-04	26	134
11854	2019-01-04	14	134

Rysunek 6 Rysunek przedstawia fragment rezultatu wykonania powyższego polecenia SELECT.

```
SELECT dWorkers, Name, Surname, Phone, Mail, Pesel FROM
mydb.Workers WHERE idWorkers =

(SELECT idManager FROM mydb.Projects WHERE  idProjects = 34);
```

Wycinek 8 Pobierz dane kierownika projektu.

idWorkers	Name	Surname	Phone	Mail	Pesel
57	Joachim	Nowacki	942764374	joachimnowacki69@158843.pl	63062287506
NULL	NULL	NULL	NULL	NULL	NULL

Rysunek 7 Rysunek przedstawia rezultat wykonania powyższego polecenia SELECT.

```
SELECT Name, DateStart, DateEnd, Budget FROM projDB.Projects

JOIN projDB.ProjFinance ON Projects.idProjects =
ProjFinance.idProjFinance;
```

Wycinek 9 Pobierz nazwę projektu, datę rozpoczęcia, datę zakończenia i budżet.

Name	DateStart	DateEnd	Budget
UpJ6W1DSZRku2KN	2015-01-02	2015-01-18	459701
TE9vzRwkl1rs552	2015-01-19	2015-03-13	381974
k2GI6k14LikWRFi	2015-03-14	2015-04-21	222166
lQBjLqUKt1aKY9p	2015-04-22	2015-05-24	142481
38vHWFit2Kc6rbp	2015-05-25	2015-07-20	141391
ZjdeeEDApi4NGBQ	2015-07-21	2015-08-15	405806
Qs5BDbjqVYE9RG1	2015-08-16	2015-10-28	400865
LxIMmpoEL7LVyZk	2015-10-29	2015-12-11	208770
2njH0v3Urb9J56s	2015-12-12	2016-02-13	234213

Rysunek 8 Rysunek przedstawia fragment rezultatu wykonania powyższego polecenia SELECT.

```
SELECT p.Name, c.CompanyName, c.NIP, c.City, c.Street,
c.ZipCode FROM projDB.Projects p

JOIN projDB.Customers c ON p.idProjects = c.idCustomers;
```

Wycinek 10 Pobierz nazwę projektu i szczegółowe informacje o klientach.

Name	CompanyName	NIP	City	Street	ZipCode
UpJ6W1DSZRku2KN	MichałSikora512	47031054841	Złoczew	Juliusza Słowackiego 873	81-330
TE9vzRwkl1rs552	LechJaniszewski488	24399422840	Łochów	Okopowa 163	37-936
k2GI6k14LikWRFi	JuliaSobolewska517	22743102288	Moryń	Garncarska 433	11-397
lQBjLqUKt1aKY9p	RozaliaWrobel537	42609108945	Nowa Sarzyna	Sienna 84	69-685
38vHWFit2Kc6rbp	AdaKowalewska512	48007975568	Morąg	Toruńska 519	64-729
ZjdeeEDApi4NGBQ	PatrycjaCzerwinska218	66476274699	Wieluń	gen. Władysława Andersa 112	12-107
Qs5BDbjqVYE9RG1	JoannaCzajkowska593	15367145337	Wojkowice	Łąkowa 378	97-610
LxIMmpoEL7LVyZk	ZanetaKołodziej597	89539911009	Ząbki	Poznańska 342	30-305
2njH0v3Urb9J56s	JanUrban41	28537326543	Leżajsk	Szewska 579	75-565
QVZlPYPO4RVZsC7	OlgaMaj344	43502137298	Ostroróg	Bolesława Limanowskiego 318	56-917

Rysunek 9 Rysunek przedstawia fragment rezultatu wykonania powyższego polecenia SELECT.

```
SELECT c.Name, c.Surname, c.CompanyName, c.NIP, p.Name FROM
mydb.Customers c
RIGHT JOIN mydb.Projects p ON p.idCustomers = c.idCustomers;
```

Wycinek 11 Pobierz klientów i nazwę projektu.

Name	Surname	CompanyName	NIP	Name
Mirosław	Laskowski	MirosławLaskowski293	72089119457	tyshQrUjUzLtXcj
Marzena	Marciniak	MarzenaMarciniak518	84712732751	JrazypDdsda8U1B
Witold	Kosiński	WitoldKosinski864	26067519484	qRgUR1ea1n4fhQC
Paweł	Bednarz	PawełBednarz132	72404701105	4iG13GF6oaI2oGD
Sylwia	Kamińska	SylwiaKaminska949	31636419596	V8Afj7hDmu31Rct
Tadeusz	Jarosz	TadeuszJarosz561	64497333882	R2fXPWGj9M4biCu

Rysunek 10 Rysunek przedstawia fragment rezultatu wykonania powyższego polecenia SELECT.

```
SELECT d.idDailyRaport, p.idProjects, p.Name, w.Surname,
w.Name FROM projDB.DailyRaport d
JOIN projDB.Projects p ON d.idProjects = p.idProjects
JOIN projDB.Workers w ON d.idWorkers = w.idWorkers
ORDER BY d.idDailyRaport, w.Surname, w.Name;
```

Wycinek 12 Pobierz id raportu, nazwę projektu, imię i nazwisko pracowników.

idDailyRaport	idProjects	Name	Surname	Name
1	2	TE9vzRwkl1rs552	Wolski	Hubert
2	2	TE9vzRwkl1rs552	Kwiatkowska	Agnieszka
3	2	TE9vzRwkl1rs552	Brzeziński	Andrzej
4	2	TE9vzRwkl1rs552	Orłowski	Zenon
5	2	TE9vzRwkl1rs552	Woźniak	Dorota
6	2	TE9vzRwkl1rs552	Kurek	Izabela

Rysunek 11 Rysunek przedstawia fragment rezultatu wykonania powyższego polecenia SELECT.

Przykładowe funkcjonalności (procedury) bazy danych.

Wszystkie procedury są dostępne pod adresem:

https://github.com/adrianpikul/projectDB/blob/main/PROCEDURES.sql

```
DROP PROCEDURE IF EXISTS mydb.addprojFinanceOne ;

DELIMITER //

CREATE PROCEDURE mydb.addprojFinanceOne

(

  IN IBudget FLOAT,

  IN IExpectedProfit FLOAT,

  IN IBonusManager FLOAT,

  IN IPenaltyPerDay FLOAT,

  IN ICosts FLOAT,

  IN IAdditionalCosts FLOAT

)

BEGIN

  INSERT INTO mydb.ProjFinance(Budget, ExpectedProfit,
BonusManager, PenaltyPerDay, Costs,  AdditionalCosts)

  VALUES (IBudget, IExpectedProfit, IBonusManager,
IPenaltyPerDay, ICosts,  IAdditionalCosts);

END //

DELIMITER ;

CALL mydb.addprojFinanceOne
(187554.2,454332.3,5642.3,1.2,56254, 1643.4);

SELECT * FROM ProjFinance ORDER BY idProjFinance  DESC;
```

Wycinek 13 Dodanie nowego rekordu do tabeli ProjFinance.

idProjFinance	Budget	ExpectedProfit	BonusManager	PenaltyPerDay	Costs	AdditionalCosts
181	187554	454332	5642.3	1.2	56254	1643.4

Rysunek 12 Rysunek przedstawia rezultat dla powyższej procedury.

```
DROP PROCEDURE IF EXISTS mydb.getAllDRForProjBetweenDate ;
DELIMITER //
CREATE PROCEDURE getAllDRForProjBetweenDate
(
    IN STARTDATE VARCHAR(10),
    IN ENDDATE VARCHAR(10),
    IN IDPROJECT INT
)
BEGIN
     SELECT * FROM mydb.DailyRaport
     WHERE Date BETWEEN STR_TO_DATE(STARTDATE,'%d,%m,%Y')
     AND STR_TO_DATE(ENDDATE,'%d,%m,%Y')
     AND DailyRaport.idProjects= IDPROJECT
     ORDER BY Date;
END //
DELIMITER ;
CALL mydb.getAllDRForProjBetweenDate ('2,1,2015', '18,1,2022',
6);
```

Wycinek 14 Procedura pobierająca wszystkie karty pracy w przedziale czasowym dla projektu.

Result Grid | Filter Rows: | Export: | Wrap Cell C

idDailyRaport	Date	idWorkers	idProjects
510	2015-02-23	50	6
511	2015-02-23	68	6
512	2015-02-23	47	6
513	2015-02-23	67	6
514	2015-02-23	6	6
515	2015-02-23	21	6
516	2015-02-24	6	6

Rysunek 13 Rysunek przedstawia fragment rezultatu dla powyższej procedury.

```
DROP PROCEDURE IF EXISTS mydb.getAllDRForWorkBetweenDate ;
DELIMITER //
CREATE PROCEDURE getAllDRForWorkBetweenDate
(
    IN STARTDATE VARCHAR(10),
    IN ENDDATE VARCHAR(10),
    IN IDWORKER INT
)
BEGIN
     SELECT * FROM DailyRaport
     WHERE Date BETWEEN STR_TO_DATE(STARTDATE,'%d,%m,%Y')
     AND STR_TO_DATE(ENDDATE,'%d,%m,%Y')
     AND idWorkers = IDWORKER;
END //
DELIMITER ;

CALL mydb.getAllDRForWorkBetweenDate ('2,1,2015', '18,1,2022',
13);
```

Wycinek 15 Procedura pobierająca wszystkie karty pracy pracownika w przedziale czasowym.

idDailyRaport	Date	idWorkers	idProjects
17	2015-01-02	13	1
45	2015-01-04	13	1
168	2015-01-14	13	1
185	2015-01-17	13	2
222	2015-01-22	13	2
308	2015-02-02	13	3
363	2015-02-08	13	4
427	2015-02-15	13	5
452	2015-02-17	13	5

Rysunek 14 Rysunek przedstawia fragment rezultatu dla powyższej procedury.

```
DROP PROCEDURE IF EXISTS mydb.updateProjFinanceFromId ;

DELIMITER //

CREATE PROCEDURE updateProjFinanceFromId

(

    IN BudgetN FLOAT,

    IN ExpcetedProfitN FLOAT,

    IN BonusManagerN FLOAT,

    IN PenaltyPerDayN FLOAT,

    IN CostsN FLOAT,

    IN AdditionalCostsN FLOAT,

    IN idProjFin INT

)

BEGIN

     UPDATE ProjFinance SET Budget = BudgetN, ExpectedProfit =
ExpcetedProfitN, BonusManager = BonusManagerN, PenaltyPerDay =
PenaltyPerDayN, Costs = CostsN, AdditionalCosts =
AdditionalCostsN WHERE idProjFinance = idProjFin;

END //

DELIMITER ;

SELECT * FROM ProjFinance where idProjFinance = 85;

CALL mydb.updateProjFinanceFromId
(187554.2,454332.3,5642.3,1.2,56254, 1643.4, 85);

SELECT * FROM ProjFinance where idProjFinance = 85;
```

Wycinek 16 Procedura aktualizująca dane rekordu w tabeli ProjFinance.

idProjFinance	Budget	ExpectedProfit	BonusManager	PenaltyPerDay	Costs	AdditionalCosts
85	187554	454332	5642.3	1.2	56254	1643.4

Rysunek 15 Rysunek przedstawia rezultat dla powyższej procedury.

```
DROP PROCEDURE IF EXISTS mydb.addConcractToIdWorker ;

DELIMITER //

CREATE PROCEDURE mydb.addConcractToIdWorker

(

  IN DateStartN VARCHAR(10),

  IN DateENDN VARCHAR(10),

  IN PositionN VARCHAR(45),

  IN RateDayN FLOAT,

  IN idWorkersN INT

)

BEGIN

  INSERT INTO mydb.Contracts(DateStart, DateEND, Position,
RateDay, idWorkers) VALUES (STR_TO_DATE(DateStartN,'%d,%m,%Y')
, STR_TO_DATE(DateENDN,'%d,%m,%Y')  , PositionN, RateDayN,
idWorkersN);

END //

DELIMITER ;

SELECT * FROM mydb.Contracts WHERE idWorkers = 90 ORDER BY
DateStart DESC;

CALL mydb.addConcractToIdWorker ('2,10,2023', '25,1,2024',
'Database specialist', 450.23, 90);

SELECT * FROM mydb.Contracts WHERE idWorkers = 90 ORDER BY
DateStart DESC;
```

Wycinek 17 Procedura dodająca nową umowę do pracownika.

idContracts	DateStart	DateEND	Position	RateDay	idWorkers
1911	2023-10-02	2024-01-25	Database specialist	450.23	90

Rysunek 16 Rysunek przedstawia rezultat dla powyższej procedury.

```
DROP PROCEDURE IF EXISTS
mydb.deleteContractidWorkersDateStartEnd ;

DELIMITER //

CREATE PROCEDURE mydb.deleteContractidWorkersDateStartEnd

(

  IN DateStartN VARCHAR(10),

  IN DateENDN VARCHAR(10),

  IN idWorkersN INT

)

BEGIN

     DELETE FROM mydb.Contracts

WHERE idWorkers = idWorkersN

AND DateStart = STR_TO_DATE(DateStartN,'%d,%m,%Y')

AND DateEND  = STR_TO_DATE(DateENDN,'%d,%m,%Y');

END //

DELIMITER ;

SELECT * FROM mydb.Contracts WHERE idWorkers = 90

ORDER BY DateStart DESC;

CALL mydb.deleteContractidWorkersDateStartEnd ('2,10,2023',
'25,1,2024', 90);

SELECT * FROM mydb.Contracts WHERE idWorkers = 90

ORDER BY DateStart DESC;
```

Wycinek 18 Procedura usuwająca umowę pracownika.

Rysunek 17 Rysunek przedstawia rezultat dla powyższej procedury.

```
DROP PROCEDURE IF EXISTS mydb.sumWorkersRentForProject ;

DELIMITER //

CREATE PROCEDURE sumWorkersRentForProject

(IN InputIdProject INT)

BEGIN

     DECLARE spDate VARCHAR(10);

     DECLARE epDate VARCHAR(10);

    SET @spDate = (SELECT DateStart FROM Projects WHERE
idProjects = InputIdProject);

    SET @epDate = (SELECT DateEnd FROM Projects WHERE
idProjects = InputIdProject);


SELECT ROUND(SUM(dd.RateDay), 2) FROM ( SELECT DISTINCT
dp.Date, dp.idWorkers, c.RateDay FROM DailyRaport dp

JOIN mydb.Contracts c ON c.idWorkers = dp.idWorkers

WHERE Date BETWEEN STR_TO_DATE(@spDate,'%Y-%m-%d')

AND STR_TO_DATE(@epDate,'%Y-%m-%d')ORDER BY dp.Date) AS dd;

END //

DELIMITER ;


CALL mydb.sumWorkersRentForProject (17);
```

Wycinek 19 Procedura wyświetlająca kwotę jaką zarobili pracownicy podczas projektu.

ROUND(SUM(dd.RateDay), 2)
36724.01

Rysunek 18 Rysunek przedstawia rezultat dla powyższej procedury.

3. Wnioski

Bazy danych to uporządkowane zbiory danych, pozwalające na przechowywanie ogromnych ilości danych. Pozwalają na płynne filtrowanie interesującej nas treści oraz szybkie porządkowanie danych. Posiadają wbudowane opcje ułatwiające wyszukiwanie interesujących nas informacji. Bazy danych są oparte na tabelach, dzięki czemu osoby, które nie mają doświadczenia są w stanie szybko pojąć strukturę bazy danych, jak i sam język SQL ze względu na jego prostotę i przyjazność użytkownikowi.

4. Spis rysunków i wycinków

Rysunek 1 Baza danych - Diagram ERD.....4
Rysunek 2 Rysunek przedstawia rezultat wykonania powyższego polecenia SELECT.....8
Rysunek 3 Rysunek przedstawia fragment rezultatu wykonania powyższego polecenia SELECT.....8
Rysunek 4 Rysunek przedstawia fragment rezultatu wykonania powyższego polecenia SELECT.....9
Rysunek 5 Rysunek przedstawia fragment rezultatu wykonania powyższego polecenia SELECT.....9
Rysunek 6 Rysunek przedstawia fragment rezultatu wykonania powyższego polecenia SELECT.....10
Rysunek 7 Rysunek przedstawia fragment rezultatu wykonania powyższego polecenia SELECT.....10
Rysunek 8 Rysunek przedstawia fragment rezultatu wykonania powyższego polecenia SELECT.....11
Rysunek 9 Rysunek przedstawia fragment rezultatu wykonania powyższego polecenia SELECT.....11
Rysunek 10 Rysunek przedstawia fragment rezultatu wykonania powyższego polecenia SELECT.....12
Rysunek 11 Rysunek przedstawia fragment rezultatu wykonania powyższego polecenia SELECT.....12
Rysunek 12 Rysunek przedstawia rezultat dla powyższej procedury.....13
Rysunek 13 Rysunek przedstawia fragment rezultatu dla powyższej procedury.....14
Rysunek 14 Rysunek przedstawia fragment rezultatu dla powyższej procedury.....15
Rysunek 15 Rysunek przedstawia fragment rezultatu dla powyższej procedury.....16
Rysunek 16 Rysunek przedstawia fragment rezultatu dla powyższej procedury.....17
Rysunek 17 Rysunek przedstawia fragment rezultatu dla powyższej procedury.....18
Rysunek 18 Rysunek przedstawia fragment rezultatu dla powyższej procedury.....19

Wycinek 1. Kod tworzący tabelę ProjFinance……6
Wycinek 2. Przykładowy kod do umieszczania danych w tabeli……8
Wycinek 3 Pobierz imię i nazwisko pracownika……8
Wycinek 4 Pobierz wszystkie projekty, gdzie powierzchnia jest większa niż 1700……8
Wycinek 5 Pobierz wszystkie umowy dla pracownika, sortuj według daty……9
Wycinek 6 Pobierz wszystkie dzienne raporty dla projektu, sortuj według id raportu……9
Wycinek 7 Pobierz wszystkie dzienne raporty pomiędzy datami.……10
Wycinek 8 Pobierz dane kierownika projektu……10
Wycinek 9 Pobierz nazwę projektu, datę rozpoczęcia, datę zakończenia i budżet……11
Wycinek 10 Pobierz nazwę projektu i szczegółowe informacje o klientach……11
Wycinek 11 Pobierz klientów i nazwę projektu……12
Wycinek 12 Pobierz id raportu, nazwę projektu, imię i nazwisko pracowników……12
Wycinek 13 Dodanie nowego rekordu do tabeli ProjFinance……13
Wycinek 14 Procedura pobierająca wszystkie karty pracy w przedziale czasowym dla projektu……14
Wycinek 15 Procedura pobierająca wszystkie karty pracy pracownika w przedziale czasowym……15
Wycinek 16 Procedura aktualizująca dane rekordu w tabeli ProjFinance……16
Wycinek 17 Procedura dodająca nową umowę do pracownika……17
Wycinek 18 Procedura usuwająca umowę pracownika……18
Wycinek 19 Procedura wyświetlająca kwotę jaką zarobili pracownicy podczas projektu……19