

Baza danych dla OSK - dokumentacja

Autor projektu: Kacper Syska

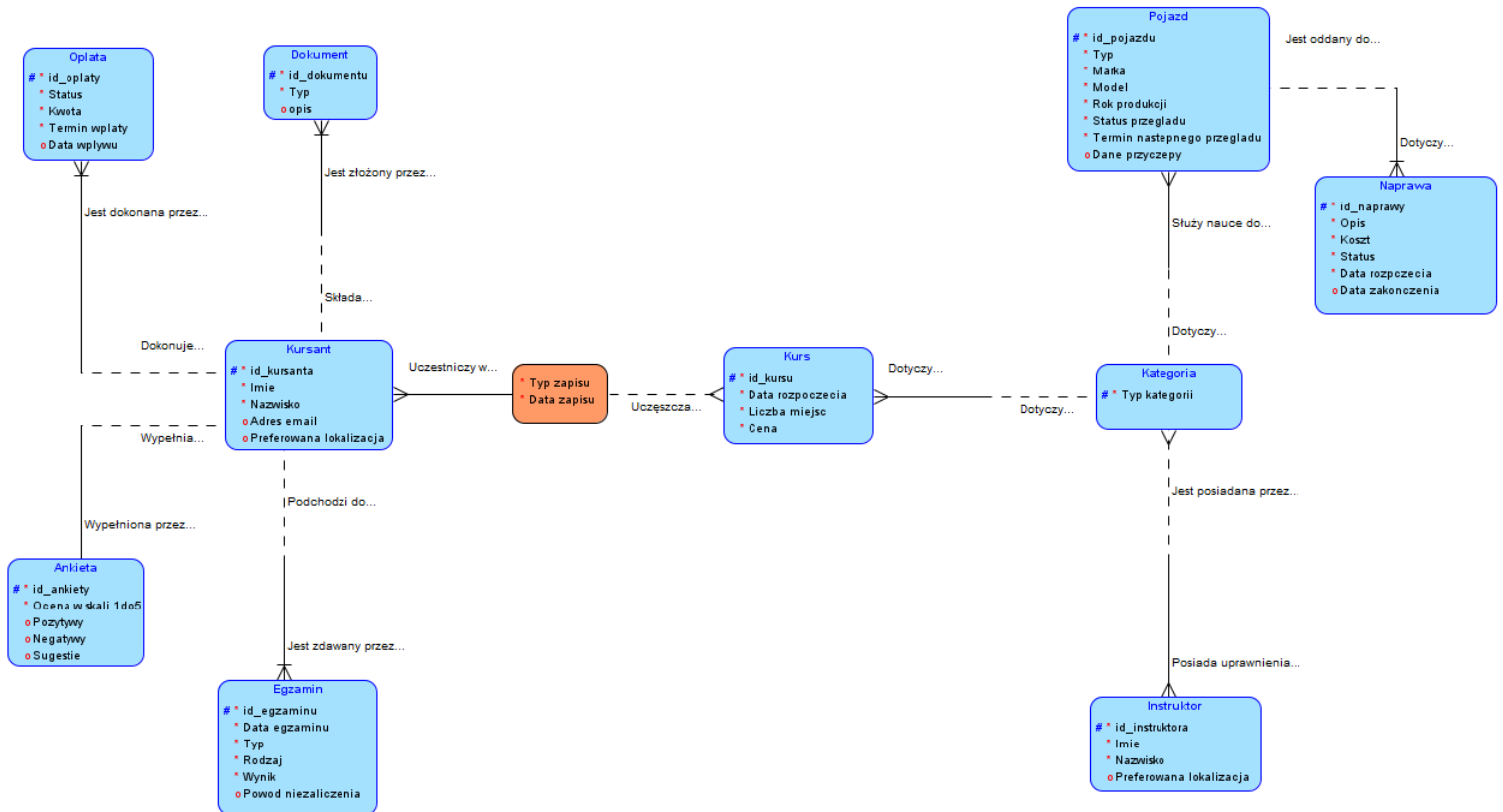
W projekcie podjęto się zaprojektowania bazy danych dla ośrodka szkolenia kierowców. Za pomocą Oracle Data Modeler opracowano model logiczny i relacyjny. Model fizyczny zaimplementowano w Oracle SQL Developer. Przygotowano przykładowy zestaw danych oraz opracowano skrypty SQL do instalacji bazy danych na serwerze, załadowania danych do tabel oraz zestaw zapytań wyszukujących konkretne statystyki i wskaźniki biznesowe. Na zestaw zapytań składają się perspektywy przykładowych użytkowników systemu tj. kursant, pracownik recepcji, zarządca pojazdów, właściciel przedsiębiorstwa. Na koniec wykonano niniejszą dokumentację projektu.

1. Analiza biznesowa projektowanej rzeczywistości.

Bazę danych powinna przechowywać:

- dane kursantów,
- dane instruktorów,
- dane samochodów,
- informacje o planowanych terminach kursów oraz o dostępnych miejscach,
- statystyki dotyczące wyników egzaminów oraz powody ewentualnego niezaliczenia,
- opinie uczestników w postaci ankiet.

2. Model logiczny i relacyjny bazy danych.



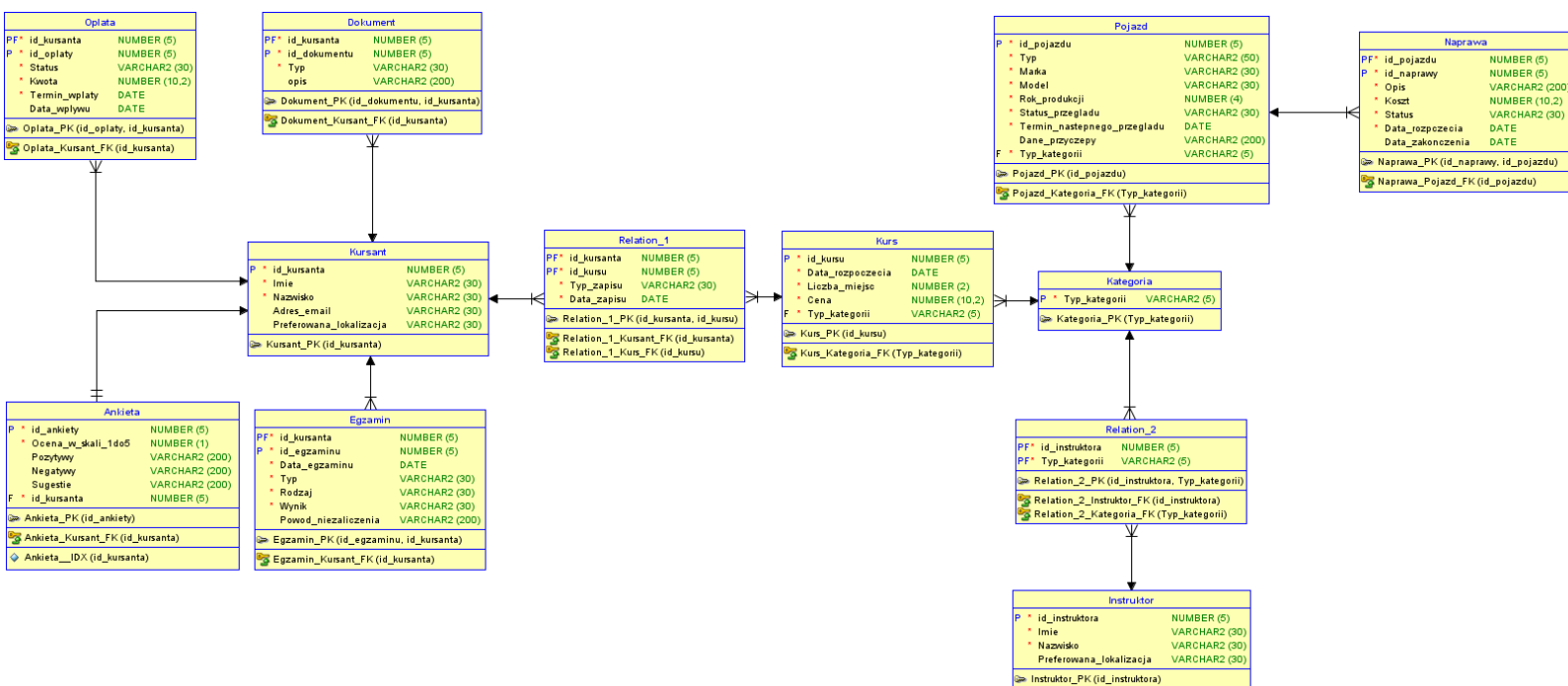
Za pomocą programu Oracle Data Modeler utworzono logiczny i relacyjny model bazy danych dla ośrodka szkolenia kierowców. Model logiczny zawiera 10 encji będących w związkach M:N i 1:N oraz opis relacji Kursant – Kurs. W modelu logicznym występują:

- elementy transakcyjne np. naprawa, opłata;
- tabela słownikowa kategoria;
- atrybuty służące do liczenia np. koszt naprawy, kwota wpłaty.

Opisy relacji pomiędzy encjami:

- **Kursant** może przesłać wiele **dokumentów**. Każdy **dokument** musi być przypisany do jednego **kursanta**.
- **Kursant** może dokonać wielu **opłat**. Każda **opłata** musi być przypisana do jednego **kursanta**.
- **Kursant** może podejść do wielu **egzaminów**. Każdy **egzamin** musi być przypisany do jednego **kursanta**.
- **Kursant** może wypełnić co najwyżej jedną **ankietę**. Każda **ankieta** musi być przypisana do jednego **kursanta**.
- **Kursant** musi być przypisany do co najmniej jednego **kursu**. Na **kurs** może być zapisanych wielu **kursantów**. Relację pomiędzy kursantem i kursem określają atrybuty: Typ zapisu (online lub na miejscu) oraz Data zapisu.
- Dany **kurs** musi dotyczyć tylko jednej **kategorii** prawa jazdy. Dana **kategoria** prawa jazdy może być realizowana na wielu **kursach**.
- **Instruktor** musi posiadać uprawnienia do przynajmniej jednej **kategorii** prawa jazdy. Tę samą **kategorię** prawa jazdy może posiadać wielu instruktorów.
- **Pojazd** musi być przypisany do jednej **kategorii**. Dana **kategoria** może być przypisana do kilku **pojazdów**.
- **Pojazd** może być wiele razy w **naprawie**. Każda **naprawa** musi dotyczyć jednego **pojazdu**.

Encje oplata, dokument, egzamin, naprawa to encje słabe (weak entity). Posiadają własne identyfikatory, natomiast ich klucze główne w modelu relacyjnym będą składać się z kombinacji ich identyfikatora oraz identyfikatora encji z którą są połączone. Dla naprawy będzie to id_pojazdu, a dla pozostałych będzie to id_kursanta. Widać to na poniższym schemacie:



Relacje M:N pomiędzy encjami Kursant – Kurs, oraz Kategoria – Instruktor zamieniły się na tabele **relation_1** i **relation_2**. Ich klucze główne składają się z kombinacji kluczy głównych tabel z którymi są w relacji.

Nowo powstała tabela **relation 1** posiada atrybuty relacji Kursant-Kurs z modelu logicznego.

3. Oprogramowanie tworzące bazę danych.

3.1. Oracle Data Modeler

- Utworzenie modelu logicznego,
- Utworzenie modelu relacyjnego,
- Utworzenie modelu fizycznego, tj. wygenerowanie skryptów SQL:
 - skrypt **create** do utworzenia bazy danych na serwerze,
 - skrypt **drop** do *de-instalacji* bazy danych z serwera.

3.2. Oracle SQL Developer

- Implementacja bazy danych na serwerze,
- Utworzenie tabel, sekwencji i perspektyw,
- Utworzenie skryptów do importu danych na podstawie przygotowanych wcześniej plików typu **.xlsx** z danymi,
- Import danych na podstawie skryptów,
- Utworzenie zdań typu **select** do użycia perspektyw i prezentacji wybranych danych.
- Możliwa de-instalacja systemu za pomocą skryptu **drop**.

3.3. LibreOffice Calc

- Utworzenie plików typu **.xlsx** oraz wprowadzenie danych do późniejszego importu.

4. Skrypty wdrożeniowe instalujące i de-instalujące zrealizowany projekt.



Utworzone skrypty można podzielić na skrypty instalujące bazę danych, skrypty wprowadzające dane do tabel (utworzone na podstawie plików .xlsx z danymi) oraz skrypt de-instalujący bazę danych. Poniżej zostały przedstawione jedynie fragmenty wybranych skryptów i plików. Wszystkie dane w plikach .xlsx, poza danymi technicznymi pojazdów i typów kategorii, zostały wymyślone na potrzeby niniejszego projektu. Wszystkie skrypty i pliki zostały w całości załączone do niniejszego dokumentu.

4.1. Skrypty instalujące baze danych.

Skrypt **OSK_DB_create.ddl** wygenerowany za pomocą Oracle Data Modeler (zmieniony później na **OSK_DB_create.sql**). Tworzy on tabele zawierające nazwy kolumn, określa typy danych w poszczególnych kolumnach, dodaje klucze główne oraz klucze obce. Poniżej zaprezentowano dwa fragmenty skryptu.

[illegible]

```

ALTER TABLE ankieta
  ADD CONSTRAINT ankieta_kursant_fk FOREIGN KEY ( id_kursanta )
    REFERENCES kursant ( id_kursanta );

ALTER TABLE dokument
  ADD CONSTRAINT dokument_kursant_fk FOREIGN KEY ( id_kursanta )
    REFERENCES kursant ( id_kursanta );

ALTER TABLE egzamin
  ADD CONSTRAINT egzamin_kursant_fk FOREIGN KEY ( id_kursanta )
    REFERENCES kursant ( id_kursanta );

ALTER TABLE kurs
  ADD CONSTRAINT kurs_kategoria_fk FOREIGN KEY ( typ_kategorii )
    REFERENCES kategoria ( typ_kategorii );

ALTER TABLE naprawa
  ADD CONSTRAINT naprawa_pojazd_fk FOREIGN KEY ( id_pojazdu )
    REFERENCES pojazd ( id_pojazdu );

ALTER TABLE oplata
  ADD CONSTRAINT oplata_kursant_fk FOREIGN KEY ( id_kursanta )
    REFERENCES kursant ( id_kursanta );

```

Skrypt **OSK_DB_seq_create.sql** tworzy w sumie 9 sekwencji oraz ustawia je jako domyślne wartości identyfikatorów tabel. Zapewnia to niepowtarzalność identyfikatorów w danej tabeli. Poniżej zaprezentowano fragment tego skryptu.

Sekwencje nie zostały utworzone dla tabel kategoria, relation_1 i relation_2. Typy kategorii, czyli identyfikatory w tabeli kategoria, są z góry ustalone. Natomiast klucze główne w tabelach relation_1 i relation_2 są kombinacjami identyfikatorów zaczerpniętych z innych tabel, a te mogą się powtarzać, np. gdy kilku kursantów jest zapisanych na ten sam kurs. Wtedy id_kursu wystąpi więcej niż raz.

```

CREATE SEQUENCE seq_id_kursu INCREMENT BY 100 START WITH 10100 MAXVALUE 99999 MINVALUE 10100 CACHE 20 ORDER NOCYCLE;

ALTER TABLE kurs MODIFY
  id_kursu DEFAULT seq_id_kursu.NEXTVAL;

-----

CREATE SEQUENCE seq_id_kursanta INCREMENT BY 1 START WITH 1000 MAXVALUE 99999 MINVALUE 1000 CACHE 20 ORDER NOCYCLE;

ALTER TABLE kursant MODIFY
  id_kursanta DEFAULT seq_id_kursanta.NEXTVAL;

-----

CREATE SEQUENCE seq_id_naprawy INCREMENT BY 1 START WITH 9001 MAXVALUE 99999 MINVALUE 9001 CACHE 20 ORDER NOCYCLE;

ALTER TABLE naprawa MODIFY
  id_naprawy DEFAULT seq_id_naprawy.NEXTVAL;

-----

CREATE SEQUENCE seq_id_oplaty INCREMENT BY 1 START WITH 8201 MAXVALUE 99999 MINVALUE 8201 CACHE 20 ORDER NOCYCLE;

ALTER TABLE oplata MODIFY
  id_oplaty DEFAULT seq_id_oplaty.NEXTVAL;

-----

CREATE SEQUENCE seq_id_pojazdu INCREMENT BY 1 START WITH 401 MAXVALUE 99999 MINVALUE 401 CACHE 20 ORDER NOCYCLE;

ALTER TABLE pojazd MODIFY
  id_pojazdu DEFAULT seq_id_pojazdu.NEXTVAL;

```

Skrypt **OSK_DB_view_create.sql** tworzy perspektywy dla kursanta z id = 1015, pracownika recepcji, kierownika ośrodka oraz nadzorcy pojazdów.

```
--perspektywa kursanta id=1015
CREATE OR REPLACE VIEW kursant1015 AS
    SELECT kursant.id_kursanta AS "Twój identyfikator", kurs.typ_kategorii AS "Kategoria", kurs.id_kursu "Identyfikator kursu",
        kurs.data_roz poczenia AS "Data rozpoczęcia kursu", oplata.kwota AS "Do opłaty", oplata.termin_wplaty AS "Termin na wpłatę",
        oplata.status as "Status opłaty", oplata.data_wplywu as "Data wpływu"
    FROM kursant, relation_1, kurs, oplata
    WHERE kursant.id_kursanta = '1015'
        AND relation_1.id_kursanta = kursant.id_kursanta
        AND relation_1.id_kursu = kurs.id_kursu
        AND oplata.id_kursanta = kursant.id_kursanta;

--perspektywa pracownika recepcji
CREATE OR REPLACE VIEW pracownik_recepcji AS
    SELECT kursant.id_kursanta, oplata.status AS status_opłaty, kurs.typ_kategorii, kurs.id_kursu, relation_1.typ_zapisu, adres_email
    FROM kursant, oplata, relation_1, kurs
    WHERE oplata.id_kursanta = kursant.id_kursanta
        AND relation_1.id_kursanta = kursant.id_kursanta
        AND relation_1.id_kursu = kurs.id_kursu;

--perspektywa kierownika ośrodka
CREATE OR REPLACE VIEW kierownik_osrodka AS
    SELECT kursant.id_kursanta, relation_1.id_kursu, typ_kategorii, ocena_w_skali_1do5, pozytywwy, negatywwy, sugestie
    FROM kursant
    FULL OUTER JOIN ankieta ON kursant.id_kursanta = ankieta.id_kursanta
    FULL OUTER JOIN relation_1 ON kursant.id_kursanta = relation_1.id_kursanta
    FULL OUTER JOIN kurs ON relation_1.id_kursu = kurs.id_kursu;

--perspektywa nadzorcy pojazdow
CREATE OR REPLACE VIEW nadzorca_pojazdow AS
    SELECT pojazd.id_pojazdu, pojazd.marka, pojazd.MODEL, pojazd.typ_kategorii, naprawa.status AS status_naprawy,
        naprawa.data_roz poczenia AS "Data rozpoczęcia naprawy", pojazd.status_przeglądu, pojazd.termin_nastepnego_przeglądu
    FROM pojazd, naprawa, kategoria
    WHERE status <> 'WYKONANA'
        AND naprawa.id_pojazdu = pojazd.id_pojazdu
        AND pojazd.typ_kategorii = kategoria.typ_kategorii
    ORDER BY termin_nastepnego_przeglądu;

COMMIT;
```

Perspektywy zgodnie z założeniami, przedstawiają wskaźniki biznesowe oraz są zbudowane w oparciu o co najmniej 3 tabele.

Warunek aby perspektywa była zbudowana w oparciu o co najmniej 3 tabele w sensowny sposób, był trudny do spełnienia dla perspektywy „nadzorca_pojazdow” w zaprojektowanym systemie.

Przy tworzeniu tej perspektywy oparto się zatem na tabeli kategoria, natomiast wartości które są w niej zawarte widnieją również w wykorzystywanej już tabeli pojazd, gdzie typ kategorii jest kluczem obcym.

Więcej istotnych wskaźników biznesowych (np. zyski w poszczególnych miesiącach, zyski ogółem) jest wyświetlanych za pomocą zapytań SELECT, które są zaprezentowane w dalszej części niniejszego dokumentu.

4.2. Wybrane skrypty wprowadzające dane do tabel.

Skrypt **KURSANT_dane.sql** wprowadza dane do utworzonej wcześniej tabeli KURSANT. Został on utworzony w SQL Developer na podstawie pliku **KURSANT_dane.xlsx** z danymi, które zostały wymyślone na potrzeby niniejszego projektu.

Skrypt nie wprowadza identyfikatora id_kursanta, ponieważ jego wartość jest domyślnie wypełniana na podstawie utworzonej wcześniej sekwencji **seq_id_kursanta**. Poniżej zaprezentowano fragment pliku .xlsx oraz fragment skryptu .sql

	A	B	C	D	E
1	ID_KURSANTA	IMIE	NAZWISKO	Adres_email	PREFEROWANA_LOKALIZACJA
2		1000 Adam	Abacki		Mokotow
3		1001 Edward	Babacki	edward.b@adres.pl	Praga
4		1002 Franciszek	Cabacki		
5		1003 Michal	Dabacki	michal.d@adres.pl	
6		1004 Jan	Ebacki		Targowek
7		1005 Ewa	Fabacka	ewa.f@adres.pl	Wola
8		1006 Aleksandra	Gabacka		
9		1007 Joanna	Habacka	joanna.h@adres.pl	
10		1008 Julia	Ibacka		Ursynow
11		1009 Wiktoria	Kabacka	wiktoria.k@adres.pl	Bielany
12		1010 Pawel	Labacka		
13		1011 Krzysztof	Mabacki	krzysztof.m@adres.pl	
14		1012 Karol	Nabacki		Ursus
15		1013 Lukasz	Obacki	lukasz.o@adres.pl	Wlochy
16		1014 Dominik	Pabacki		
17		1015 Klaudia	Rabacki	klaudia.r@adres.pl	
18		1016 Katarzyna	Sabacka		Goclaw
19		1017 Zofia	Tabacka	zofia.t@adres.pl	Zoliborz
20		1018 Kamila	Wabacka		
21		1019 Marta	Zabacka	marta.z@adres.pl	
22		1020 Adam	Bebacki		Bialoleka
23		1021 Edward	Cebacki		Srodmiescie
24		1022 Franciszek	Debacki	franiszek.d@adres.pl	
25		1023 Michal	Febacki		

```
SET DEFINE OFF
```

```
INSERT INTO KURSANT (IMIE, NAZWISKO, ADRES_EMAIL, PREFEROWANA_LOKALIZACJA)
VALUES ('Adam', 'Abacki', NULL, 'Mokotow');
```

```
INSERT INTO KURSANT (IMIE, NAZWISKO, ADRES_EMAIL, PREFEROWANA_LOKALIZACJA)
VALUES ('Edward', 'Babacki', 'edward.b@adres.pl', 'Praga');
```

```
INSERT INTO KURSANT (IMIE, NAZWISKO, ADRES_EMAIL, PREFEROWANA_LOKALIZACJA)
VALUES ('Franciszek', 'Cabacki', NULL, NULL);
```

```
INSERT INTO KURSANT (IMIE, NAZWISKO, ADRES_EMAIL, PREFEROWANA_LOKALIZACJA)
VALUES ('Michal', 'Dabacki', 'michal.d@adres.pl', NULL);
```

```
INSERT INTO KURSANT (IMIE, NAZWISKO, ADRES_EMAIL, PREFEROWANA_LOKALIZACJA)
VALUES ('Jan', 'Ebacki', NULL, 'Targowek');
```

```
INSERT INTO KURSANT (IMIE, NAZWISKO, ADRES_EMAIL, PREFEROWANA_LOKALIZACJA)
VALUES ('Ewa', 'Fabacka', 'ewa.f@adres.pl', 'Wola');
```

```
INSERT INTO KURSANT (IMIE, NAZWISKO, ADRES_EMAIL, PREFEROWANA_LOKALIZACJA)
VALUES ('Aleksandra', 'Gabacka', NULL, NULL);
```


Skrypt **POJAZD_dane.sql** wprowadza dane do utworzonej wcześniej tabeli POJAZD. Został on utworzony w SQL Developer na podstawie pliku **POJAZD_dane.xlsx** z danymi, które poza danymi technicznymi i typami kategorii, zostały wymyślone na potrzeby niniejszego projektu. Poniżej zaprezentowano fragment pliku .xlsx oraz fragment skryptu .sql.

Skrypt nie wprowadza identyfikatora id_naprawy, ponieważ jego wartość jest domyślnie wypełniana na podstawie utworzonej wcześniej sekwencji seq_id_pojazdu.

	A	B	C	D	E	F	G	H	I
1	ID_POJAZDU	TYP	MARKA	MODEL	ROK_PRODUKCJI	STATUS_PRZEGladU	TERMIN_NASTEPNEGO_PRZEGladU	DANE_PRZYCZEPY	TYP_KATEGORII
2		401 Samochod osobowy	Hyundai	i20	2018	AKTUALNY	12.01.2025		B
3		402 Samochod osobowy	Hyundai	i20	2019	AKTUALNY	10.02.2025		B
4		403 Samochod osobowy	Hyundai	i20	2019	AKTUALNY	4.06.2024		B
5		404 Samochod osobowy	Hyundai	i20	2021	NIEAKTUALNY	25.01.2024		B
6		405 Samochod osobowy	Hyundai	i20	2022	AKTUALNY	12.10.2024		B
7		406 Samochod osobowy	Hyundai	i20	2020	AKTUALNY	8.04.2024		B
8		407 Samochod osobowy	Hyundai	i20	2020	AKTUALNY	14.08.2024		B
9		408 Samochod osobowy	Hyundai	i20	2019	NIEAKTUALNY	22.01.2024		B
10		409 Motocykl	Suzuki	SV 650	2019	AKTUALNY	4.02.2025		A
11		410 Motocykl	Suzuki	SV 650	2022	AKTUALNY	12.01.2025		A
12		411 Motocykl	Suzuki	SFV 650UA 35kW	2021	AKTUALNY	14.08.2024		A2
13		412 Motocykl	Suzuki	SFV 650UA 35kW	2021	AKTUALNY	8.04.2024		A2
14		413 Motocykl	Suzuki	SFV 650UA 35kW	2020	NIEAKTUALNY	18.01.2024		A2
15		414 Motocykl	Honda	CBF 125M	2022	AKTUALNY	8.04.2024		A1
16		415 Motorower	Romet	Zetka 50	2020	AKTUALNY	4.06.2024		AM
17		416 Motorower	Kymco	Agility 50	2019	AKTUALNY	10.02.2025		AM
18		417 Samochod	Renault	Trafic	2018	AKTUALNY	12.01.2025	przyczepa Niewiadów B1300	BE
19		418 Samochod ciężarowy	MAN	TGL 12.220	2019	AKTUALNY	4.02.2025		C
20		419 Samochod ciężarowy	MAN	TGL 12.250	2020	AKTUALNY	10.02.2025		C
21		420 Samochod ciężarowy	DAF	LF	2019	AKTUALNY	4.06.2024		C
22		421 Samochod ciężarowy	MAN	TGL 12	2017	NIEAKTUALNY	5.02.2024	przyczepa typu tandem	CE
23		422 Samochod ciężarowy	DAF	LF	2018	AKTUALNY	8.04.2024	przyczepa typu tandem	CE
24		423 Autobus	Solbus	ST 11	2019	AKTUALNY	12.01.2025		D
25		424 Ciągnik rolniczy	NEW HOLLAND	T4.65	2020	AKTUALNY	14.08.2024	przyczepa PRONAR T663/2	T
26		425 Ciągnik rolniczy	ZETOR	PROXIMA PLUS 80	2019	AKTUALNY	8.04.2024	przyczepa MAR-POL MT 600	T

SET DEFINE OFF

```
INSERT INTO POJAZD (TYP, MARKA, MODEL, ROK_PRODUKCJI, STATUS_PRZEGladU, TERMIN_NASTEPNEGO_PRZEGladU, DANE_PRZYCZEPY, TYP_KATEGORII)
VALUES ('Samochod osobowy', 'Hyundai', 'i20', 2018.0, 'AKTUALNY', to_date('12/01/2025', 'DD/MM/YYYY'), NULL, 'B');

INSERT INTO POJAZD (TYP, MARKA, MODEL, ROK_PRODUKCJI, STATUS_PRZEGladU, TERMIN_NASTEPNEGO_PRZEGladU, DANE_PRZYCZEPY, TYP_KATEGORII)
VALUES ('Samochod osobowy', 'Hyundai', 'i20', 2019.0, 'AKTUALNY', to_date('10/02/2025', 'DD/MM/YYYY'), NULL, 'B');

INSERT INTO POJAZD (TYP, MARKA, MODEL, ROK_PRODUKCJI, STATUS_PRZEGladU, TERMIN_NASTEPNEGO_PRZEGladU, DANE_PRZYCZEPY, TYP_KATEGORII)
VALUES ('Samochod osobowy', 'Hyundai', 'i20', 2019.0, 'AKTUALNY', to_date('4/06/2024', 'DD/MM/YYYY'), NULL, 'B');

INSERT INTO POJAZD (TYP, MARKA, MODEL, ROK_PRODUKCJI, STATUS_PRZEGladU, TERMIN_NASTEPNEGO_PRZEGladU, DANE_PRZYCZEPY, TYP_KATEGORII)
VALUES ('Samochod osobowy', 'Hyundai', 'i20', 2021.0, 'NIEAKTUALNY', to_date('25/01/2024', 'DD/MM/YYYY'), NULL, 'B');

INSERT INTO POJAZD (TYP, MARKA, MODEL, ROK_PRODUKCJI, STATUS_PRZEGladU, TERMIN_NASTEPNEGO_PRZEGladU, DANE_PRZYCZEPY, TYP_KATEGORII)
VALUES ('Samochod osobowy', 'Hyundai', 'i20', 2022.0, 'AKTUALNY', to_date('12/10/2024', 'DD/MM/YYYY'), NULL, 'B');

INSERT INTO POJAZD (TYP, MARKA, MODEL, ROK_PRODUKCJI, STATUS_PRZEGladU, TERMIN_NASTEPNEGO_PRZEGladU, DANE_PRZYCZEPY, TYP_KATEGORII)
VALUES ('Samochod osobowy', 'Hyundai', 'i20', 2020.0, 'AKTUALNY', to_date('8/04/2024', 'DD/MM/YYYY'), NULL, 'B');

INSERT INTO POJAZD (TYP, MARKA, MODEL, ROK_PRODUKCJI, STATUS_PRZEGladU, TERMIN_NASTEPNEGO_PRZEGladU, DANE_PRZYCZEPY, TYP_KATEGORII)
VALUES ('Samochod osobowy', 'Hyundai', 'i20', 2020.0, 'AKTUALNY', to_date('14/08/2024', 'DD/MM/YYYY'), NULL, 'B');

INSERT INTO POJAZD (TYP, MARKA, MODEL, ROK_PRODUKCJI, STATUS_PRZEGladU, TERMIN_NASTEPNEGO_PRZEGladU, DANE_PRZYCZEPY, TYP_KATEGORII)
VALUES ('Samochod osobowy', 'Hyundai', 'i20', 2019.0, 'NIEAKTUALNY', to_date('22/01/2024', 'DD/MM/YYYY'), NULL, 'B');
```

Skrypt **NAPRAWA_dane.sql** wprowadza dane do utworzonej wcześniej tabeli **NAPRAWA**. Został on utworzony w SQL Developer na podstawie pliku **NAPRAWA_dane.xlsx** z danymi, które zostały wymyślane na potrzeby niniejszego projektu. Poniżej zaprezentowano fragment pliku .xlsx oraz fragment skryptu .sql. Skrypt nie wprowadza identyfikatora **id_naprawy**, ponieważ jego wartość jest domyślnie wypełniana na podstawie utworzonej wcześniej sekwencji **seq_id_naprawy**.

Warto zaznaczyć że **id_pojazdu** tym razem jest wprowadzane, tak jak pozostałe wartości danych. W tym przypadku nie wykorzystano sekwencji, ponieważ ten sam pojazd może być poddany wielu naprawom. Innymi słowy, różne naprawy, czyli unikalne wartości **id_naprawa**, mogą dotyczyć tego samego pojazdu, czyli tej samej wartości **id_pojazdu**. Taka sama sytuacja występuje w skryptach wprowadzających dane do tabel **egzamin**, **opłata**, **ankieta**, **egzamin**.

	A	B	C	D	E	F	G
1	ID_POJAZDU	ID_NAPRAWY	OPIS	KOSZT	STATUS	DATA_ROZPOCZECIA	DATA_ZAKONCZENIA
2	401	9001	Wymiana klockow hamulocywch	500	WYKONANA	5.01.2024	10.01.2024
3	425	9002	Wymiana tarcz	1000	WYKONANA	11.01.2024	14.01.2024
4	402	9003	Wymiana oleju i filtrow	300	WYKONANA	15.01.2024	19.01.2024
5	424	9004	Wymiana pompy paliwa	1200	WYKONANA	20.01.2024	24.01.2024
6	403	9005	Wymiana opon	250	W TRAKCIE	25.01.2024	
7	423	9006	Wymiana swiec zaplonowych	900	WYKONANA	30.01.2024	5.02.2024
8	404	9007	Montaz nowej rury wydechowej	400	WYKONANA	10.02.2024	15.02.2024
9	422	9008	Wymiana katalizatora	1500	WYKONANA	16.02.2024	20.02.2024
10	405	9009	Wymiana cewki	300	WYKONANA	21.02.2024	25.02.2024
11	421	9010	Wymiana szby – przednia prawa	400	W TRAKCIE	26.02.2024	
12	406	9011	Wymiana wahacza	650	DO ODBIOR	28.02.2024	2.03.2024
13	420	9012	Wymiana miski olejowej	300	W TRAKCIE	1.03.2024	
14	407	9013	Wymiana filtrow powietrza	80	W TRAKCIE	4.03.2024	
15	419	9014	Serwis klimatyzacji	200	WYKONANA	10.12.2023	15.12.2023
16	408	9015	Wymiana pompy paliwa	1200	WYKONANA	5.12.2023	9.12.2023
17	418	9016	Wymiana oleju i filtrow	300	W TRAKCIE	26.02.2023	
18	409	9017	Wymiana swiec zaplonowych	900	WYKONANA	10.10.2023	12.10.2023
19	417	9018	Wymiana opon	250	WYKONANA	15.10.2023	20.10.2023
20	410	9019	Wymiana oleju	300	WYKONANA	21.10.2023	24.10.2023
21	416	9020	Wymiana klockow hamulocywch	500	WYKONANA	25.10.2023	27.10.2023
22	411	9021	Wymiana klockow hamulocywch	500	WYKONANA	6.01.2024	11.01.2024
23	415	9022	Wymiana tarcz	1000	WYKONANA	12.01.2024	15.01.2024
24	412	9023	Wymiana oleju	300	DO ODBIOR	16.01.2024	20.01.2024
25	414	9024	Wymiana pompy paliwa	1200	W TRAKCIE	21.01.2024	

```
SET DEFINE OFF
```

```
INSERT INTO NAPRAWA (ID_POJAZDU, OPIS, KOSZT, STATUS, DATA_ROZPCZECIA, DATA_ZAKONCZENIA)
VALUES (401.0, 'Wymiana klockow hamulocywch', 500.0, 'WYKONANA', to_date('5/01/2024', 'DD/MM/YYYY'), to_date('10/01/2024', 'DD/MM/YYYY'));

INSERT INTO NAPRAWA (ID_POJAZDU, OPIS, KOSZT, STATUS, DATA_ROZPCZECIA, DATA_ZAKONCZENIA)
VALUES (425.0, 'Wymiana tarcz', 1000.0, 'WYKONANA', to_date('11/01/2024', 'DD/MM/YYYY'), to_date('14/01/2024', 'DD/MM/YYYY'));

INSERT INTO NAPRAWA (ID_POJAZDU, OPIS, KOSZT, STATUS, DATA_ROZPCZECIA, DATA_ZAKONCZENIA)
VALUES (402.0, 'Wymiana oleju i filtrow', 300.0, 'WYKONANA', to_date('15/01/2024', 'DD/MM/YYYY'), to_date('19/01/2024', 'DD/MM/YYYY'));

INSERT INTO NAPRAWA (ID_POJAZDU, OPIS, KOSZT, STATUS, DATA_ROZPCZECIA, DATA_ZAKONCZENIA)
VALUES (424.0, 'Wymiana pompy paliwa', 1200.0, 'WYKONANA', to_date('20/01/2024', 'DD/MM/YYYY'), to_date('24/01/2024', 'DD/MM/YYYY'));

INSERT INTO NAPRAWA (ID_POJAZDU, OPIS, KOSZT, STATUS, DATA_ROZPCZECIA, DATA_ZAKONCZENIA)
VALUES (403.0, 'Wymiana opon', 250.0, 'W TRAKCIE', to_date('25/01/2024', 'DD/MM/YYYY'), NULL);

INSERT INTO NAPRAWA (ID_POJAZDU, OPIS, KOSZT, STATUS, DATA_ROZPCZECIA, DATA_ZAKONCZENIA)
VALUES (423.0, 'Wymiana swiec zaplonowych', 900.0, 'WYKONANA', to_date('30/01/2024', 'DD/MM/YYYY'), to_date('5/02/2024', 'DD/MM/YYYY'));

INSERT INTO NAPRAWA (ID_POJAZDU, OPIS, KOSZT, STATUS, DATA_ROZPCZECIA, DATA_ZAKONCZENIA)
VALUES (404.0, 'Montaz nowej rury wydechowej', 400.0, 'WYKONANA', to_date('10/02/2024', 'DD/MM/YYYY'), to_date('15/02/2024', 'DD/MM/YYYY'));

INSERT INTO NAPRAWA (ID_POJAZDU, OPIS, KOSZT, STATUS, DATA_ROZPCZECIA, DATA_ZAKONCZENIA)
VALUES (422.0, 'Wymiana katalizatora', 1500.0, 'WYKONANA', to_date('16/02/2024', 'DD/MM/YYYY'), to_date('20/02/2024', 'DD/MM/YYYY'));

INSERT INTO NAPRAWA (ID_POJAZDU, OPIS, KOSZT, STATUS, DATA_ROZPCZECIA, DATA_ZAKONCZENIA)
VALUES (405.0, 'Wymiana cewki', 300.0, 'WYKONANA', to_date('21/02/2024', 'DD/MM/YYYY'), to_date('25/02/2024', 'DD/MM/YYYY'));
```

4.3. Skrypt de-instalujący bazę danych

Skrypt **OSK_DB_drop.sql** de-instaluje wszystkie utworzone wcześniej sekwencje, perspektywy i tabele, łącznie z wprowadzonymi do nich wcześniej danymi.

Został on utworzony na podstawie skryptu **OSK_DB_drop.ddl** wygenerowanego za pomocą Oracle Data Modeler, poprzez dodanie poleceń dotyczących sekwencji i perspektyw, dodanie frazy COMMIT oraz zapisanie w formacie .sql.

```
DROP TABLE ANKIETA CASCADE CONSTRAINTS;

DROP TABLE DOKUMENT CASCADE CONSTRAINTS;

DROP TABLE EGZAMIN CASCADE CONSTRAINTS;

DROP TABLE INSTRUKTOR CASCADE CONSTRAINTS;

DROP TABLE KATEGORIA CASCADE CONSTRAINTS;

DROP TABLE KURS CASCADE CONSTRAINTS;

DROP TABLE KURSANT CASCADE CONSTRAINTS;

DROP TABLE NAPRAWA CASCADE CONSTRAINTS;

DROP TABLE OPLATA CASCADE CONSTRAINTS;

DROP TABLE POJAZD CASCADE CONSTRAINTS;

DROP TABLE RELATION_1 CASCADE CONSTRAINTS;

DROP TABLE RELATION_2 CASCADE CONSTRAINTS;

DROP SEQUENCE SEQ_ID_ANKIETY;

DROP SEQUENCE SEQ_ID_DOKUMENTU;

DROP SEQUENCE SEQ_ID_EGZAMINU;

DROP SEQUENCE SEQ_ID_INSTRUKTORA;

DROP SEQUENCE SEQ_ID_KURSU;

DROP SEQUENCE SEQ_ID_KURSANTA;

DROP SEQUENCE SEQ_ID_NAPRAWY;

DROP SEQUENCE SEQ_ID_OPLATY;

DROP SEQUENCE SEQ_ID_POJAZDU;

DROP VIEW KURSANT1015;

DROP VIEW PRACOWNIK_RECEPCJI;

DROP VIEW KIEROWNIK_OSRODKA;
```

5. Instrukcja instalacji projektu i sprawdzenia jego poprawności.

Projekt został w całości zaimplementowany na serwerze w programie Oracle SQL Developer. Do niniejszej dokumentacji załączono folder ze wszystkimi wymienionymi plikami i skryptami.

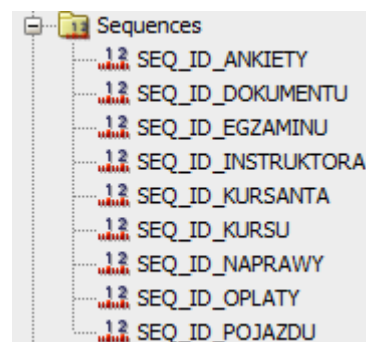
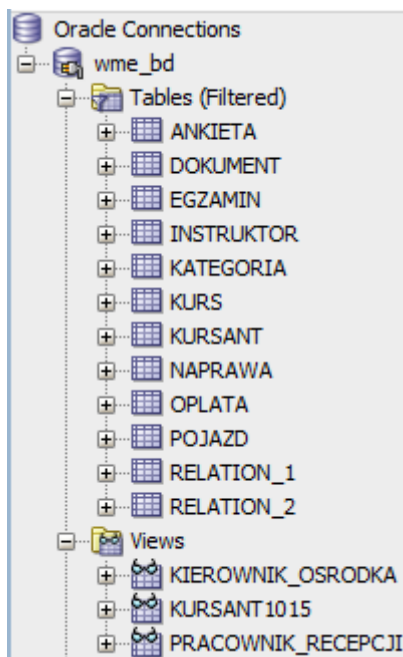
W celu zainstalowania projektu na serwerze bazodanowym, należy za pomocą programu SQL Developer połączyć się z wybranym serwerem, otworzyć wszystkie zamieszczone skrypty, uruchomić je jeden po drugim **według ustalonej kolejności**. **Nie należy wgrywać skryptu nr 17.** Jest to skrypt de-instalujący całą bazę danych – perspektywy, sekwencje, tabele wraz z danymi.

UWAGA! Ważne jest, aby skrypty wgrywać według ustalonej kolejności. Jeśli kolejność nie zostanie zachowana, może wystąpić błąd o naruszeniu więzów spójności¹.

5.1. Skrypty „create”

Po wgraniu pierwszych 3 skryptów create, w pasku bocznym powinny się wyświetlić ikony tabel, perspektyw i sekwencji (Tables, Views, Sequences). Jeśli ich nie widać, należy odświeżyć widok klikając w ikonę z niebieskimi strzałkami o nazwie refresh. Pojawienie się tych ikon świadczy o poprawnym utworzeniu tabel, perspektyw i sekwencji.

- 01.OSK_DB_create
- 02.OSK_DB_seq_create
- 03.OSK_DB_view_create
- 04.KURSANT_dane
- 05.KATEGORIA_dane
- 06.INSTRUKTOR_dane
- 07.KURS_dane
- 08.POJAZD_dane
- 09.NAPRAWA_dane
- 10.DOKUMENT_dane
- 11.OPLATA_dane
- 12.EGZAMIN_dane
- 13.ANKIETA_dane
- 14.RELATION_1_dane
- 15.RELATION_2_dane
- 16.SELECT_statements
- 17.OSK_DB_drop



¹ Jeśli przy wgrywaniu danych np. do tabeli ankieta, tabela kursant będzie pusta, to wystąpi błąd. Kluczem głównym tabeli ankieta jest kombinacja identyfikatorów id_ankiety i id_kursant, zatem jest on uzależniony od wartości w tabeli kursant. Jest to zgodne z logiką ustaloną przy projektowaniu tej bazy danych: kursant może wypełnić co najwyżej jedną ankietę, natomiast każda ankieta musi być przypisana do jednego kursanta.

Wystąpienie błędu o naruszeniu więzów spójności może oznaczać że zaszła próba wgrania danych np. do tabeli ANKIETA, bez wcześniejszego wgrania danych do tabeli kursant.

Po kliknięciu w konkretną tabelę, powinny być widoczne informacje nt. tej tabeli np. nazwy kolumn, typu danych, opcjonalność (nullable) oraz utworzone wcześniej sekwencje. Na poniższych przykładach widać, że domyślnymi wartościami identyfikatorów są wartości sekwencji utworzone wcześniej.

Tabela „Kursant”

❖ COLUMN_NAME	❖ DATA_TYPE	❖ NULLABLE	DATA_DEFAULT	❖ COLUMN_ID	❖ COMMENTS
1 ID_KURSANTA	NUMBER(5,0)	No	"WME21BC1Z_06"."SEQ_ID_KURSANTA"."NEXTVAL"	1	(null)
2 IMIE	VARCHAR2(30 BYTE)	No	(null)	2	(null)
3 NAZWISKO	VARCHAR2(30 BYTE)	No	(null)	3	(null)
4 ADRES_EMAIL	VARCHAR2(30 BYTE)	Yes	(null)	4	(null)
5 PREFEROWANA_LOKALIZACJA	VARCHAR2(30 BYTE)	Yes	(null)	5	(null)

Tabela „Pojazd”

❖ COLUMN_NAME	❖ DATA_TYPE	❖ NULLABLE	DATA_DEFAULT	❖ COLUMN_ID	❖ COMMENTS
1 ID_POJAZDU	NUMBER(5,0)	No	"WME21BC1Z_06"."SEQ_ID_POJAZDU"."NEXTVAL"	1	(null)
2 TYP	VARCHAR2(50 BYTE)	No	(null)	2	(null)
3 MARKA	VARCHAR2(30 BYTE)	No	(null)	3	(null)
4 MODEL	VARCHAR2(30 BYTE)	No	(null)	4	(null)
5 ROK_PRODUKCJI	NUMBER(4,0)	No	(null)	5	(null)
6 STATUS_PRZEGLADU	VARCHAR2(30 BYTE)	No	(null)	6	(null)
7 TERMIN_NASTEPNEGO_PRZEGLADU	DATE	No	(null)	7	(null)
8 DANE_PRZYCZEPY	VARCHAR2(200 BYTE)	Yes	(null)	8	(null)
9 TYP_KATEGORII	VARCHAR2(5 BYTE)	No	(null)	9	(null)

Tabela „Naprawa”

❖ COLUMN_NAME	❖ DATA_TYPE	❖ NULLABLE	DATA_DEFAULT	❖ COLUMN_ID	❖ COMMENTS
1 ID_POJAZDU	NUMBER(5,0)	No	(null)	1	(null)
2 ID_NAPRAWY	NUMBER(5,0)	No	"WME21BC1Z_06"."SEQ_ID_NAPRAWY"."NEXTVAL"	2	(null)
3 OPIS	VARCHAR2(200 BYTE)	No	(null)	3	(null)
4 KOSZT	NUMBER(10,2)	No	(null)	4	(null)
5 STATUS	VARCHAR2(30 BYTE)	No	(null)	5	(null)
6 DATA_ROZPCZECIA	DATE	No	(null)	6	(null)
7 DATA_ZAKONCZENIA	DATE	Yes	(null)	7	(null)

Po sprawdzeniu poprawności utworzenia tabel, perspektyw i sekwencji można przejść do wprowadzenia danych.

5.2. Skrypty z danymi

Aby wprowadzić dane do tabel, należy wgrać po kolei kolejne skrypty o numerach od 4 do 15. Jeśli nie wystąpiły komunikaty o błędach, to wszystkie skrypty zostały poprawnie załadowane. Można to sprawdzić klikając w poszczególne tabele, a następnie przejść do zakładki „Data”. W kolumnach powinny być widoczne wprowadzone dane, co obrazują poniższe przykłady.

Jeśli dane nie są widoczne, należy odświeżyć widok klikając w ikonę z niebieskimi strzałkami o nazwie refresh.

Fragment danych wgranych do tabeli „Kursant”

ID_KURSANTA	IMIE	NAZWISKO	ADRES_EMAIL	PREFEROWANA_LOKALIZACJA
1	1000 Adam	Abacki	(null)	Mokotow
2	1001 Edward	Babacki	edward.b@adres.pl	Praga
3	1002 Franciszek	Cabacki	(null)	(null)
4	1003 Michal	Dabacki	michal.d@adres.pl	(null)
5	1004 Jan	Ebacki	(null)	Targowek
6	1005 Ewa	Fabacka	ewa.f@adres.pl	Wola
7	1006 Aleksandra	Gabacka	(null)	(null)
8	1007 Joanna	Habacka	joanna.h@adres.pl	(null)
9	1008 Julia	Ibacka	(null)	Ursynow
10	1009 Wiktoria	Kabacka	wiktoria.k@adres.pl	Bielany
11	1010 Pawel	Labacka	(null)	(null)
12	1011 Krzysztof	Mabacki	krzysztof.m@adres.pl	(null)
13	1012 Karol	Nabacki	(null)	Ursus
14	1013 Lukasz	Obacki	lukasz.o@adres.pl	Wlochy
15	1014 Dominik	Pabacki	(null)	(null)
16	1015 Klaudia	Rabacki	klaudia.r@adres.pl	(null)
17	1016 Katarzyna	Sabacka	(null)	Goclaw
18	1017 Zofia	Tabacka	zofia.t@adres.pl	Zoliborz
19	1018 Kamila	Wabacka	(null)	(null)
20	1019 Marta	Zabacka	marta.z@adres.pl	(null)

Dane wgrane do tabeli „Kursant”

ID_POJAZDU	TYP	MARKA	MODEL	ROK_PRODUKCJI	STATUS_PRZEGladu	TERMIN_NASTEPNEGO_PRZEGladu	DANE_PRZYCZEPY	TYP_KATEGORII
1	401 Samochod osobowy	Hyundai	120	2018	AKTUALNY	25/01/12	(null)	B
2	402 Samochod osobowy	Hyundai	120	2019	AKTUALNY	25/02/10	(null)	B
3	403 Samochod osobowy	Hyundai	120	2019	AKTUALNY	24/06/04	(null)	B
4	404 Samochod osobowy	Hyundai	120	2021	NIEAKTUALNY	24/01/25	(null)	B
5	405 Samochod osobowy	Hyundai	120	2022	AKTUALNY	24/10/12	(null)	B
6	406 Samochod osobowy	Hyundai	120	2020	AKTUALNY	24/04/08	(null)	B
7	407 Samochod osobowy	Hyundai	120	2020	AKTUALNY	24/08/14	(null)	B
8	408 Samochod osobowy	Hyundai	120	2019	NIEAKTUALNY	24/01/22	(null)	B
9	409 Motocykl	Suzuki	SV 650	2019	AKTUALNY	25/02/04	(null)	A
10	410 Motocykl	Suzuki	SV 650	2022	AKTUALNY	25/01/12	(null)	A
11	411 Motocykl	Suzuki	SVF 650UA 35kW	2021	AKTUALNY	24/08/14	(null)	A2
12	412 Motocykl	Suzuki	SVF 650UA 35kW	2021	AKTUALNY	24/04/08	(null)	A2
13	413 Motocykl	Suzuki	SVF 650UA 35kW	2020	NIEAKTUALNY	24/01/18	(null)	A2
14	414 Motocykl	Honda	CBF 125M	2022	AKTUALNY	24/04/08	(null)	A1
15	415 Motorower	Romet	Zetka 50	2020	AKTUALNY	24/06/04	(null)	AM
16	416 Motorower	Kymco	Agility 50	2019	AKTUALNY	25/02/10	(null)	AM
17	417 Samochod	Renault	Trafic	2018	AKTUALNY	25/01/12	przyczepa Niewiadom B1300 BE	BE
18	418 Samochod ciezarowy	MAN	TGL 12.220	2019	AKTUALNY	25/02/04	(null)	C
19	419 Samochod ciezarowy	MAN	TGL 12.250	2020	AKTUALNY	25/02/10	(null)	C
20	420 Samochod ciezarowy	DAF	LF	2019	AKTUALNY	24/06/04	(null)	C
21	421 Samochod ciezarowy	MAN	TGL 12	2017	NIEAKTUALNY	24/02/05	przyczepa typu tandem	CE
22	422 Samochod ciezarowy	DAF	LF	2018	AKTUALNY	24/04/08	przyczepa typu tandem	CE
23	423 Autobus	Solbus	ST 11	2019	AKTUALNY	25/01/12	(null)	D
24	424 Ciagnik rolniczy	NEW HOLLAND	T4.65	2020	AKTUALNY	24/08/14	przyczepa FRONAR T663/2	T
25	425 Ciagnik rolniczy	ZETOR	PROXIMA PLUS 80	2019	AKTUALNY	24/04/08	przyczepa MAR-POL MT 600	T

Fragment danych wgranych do tabeli „Naprawa”

ID_POJAZDU	ID_NAPRAWY	OPIS	KOSZT	STATUS	DATA_ROZPCZECIA	DATA_ZAKONCZENIA
1	401	9001 Wymiana klocek hamulcowych	500	WYKONANA	24/01/05	24/01/10
2	425	9002 Wymiana tarcz	1000	WYKONANA	24/01/11	24/01/14
3	402	9003 Wymiana oleju i filtrow	300	WYKONANA	24/01/15	24/01/19
4	424	9004 Wymiana pompy paliwa	1200	WYKONANA	24/01/20	24/01/24
5	403	9005 Wymiana opon	250	W TRAKCIE	24/01/25	(null)
6	423	9006 Wymiana swiec zaplonowych	900	WYKONANA	24/01/30	24/02/05
7	404	9007 Montaz nowej rury wydechowej	400	WYKONANA	24/02/10	24/02/15
8	422	9008 Wymiana katalizatora	1500	WYKONANA	24/02/16	24/02/20
9	405	9009 Wymiana cewki	300	WYKONANA	24/02/21	24/02/25
10	421	9010 Wymiana szby - przednia prawa	400	W TRAKCIE	24/02/26	(null)
11	406	9011 Wymiana wahacza	650	DO ODBIORU	24/02/28	24/03/02
12	420	9012 Wymiana miski olejowej	300	W TRAKCIE	24/03/01	(null)
13	407	9013 Wymiana filtrow powietrza	80	W TRAKCIE	24/03/04	(null)
14	419	9014 Serwis klimatyzacji	200	WYKONANA	23/12/10	23/12/15
15	408	9015 Wymiana pompy paliwa	1200	WYKONANA	23/12/05	23/12/09
16	418	9016 Wymiana oleju i filtrow	300	W TRAKCIE	23/02/26	(null)
17	409	9017 Wymiana swiec zaplonowych	900	WYKONANA	23/10/10	23/10/12
18	417	9018 Wymiana opon	250	WYKONANA	23/10/15	23/10/20
19	410	9019 Wymiana oleju	300	WYKONANA	23/10/21	23/10/24
20	416	9020 Wymiana klocek hamulcowych	500	WYKONANA	23/10/25	23/10/27

5.3. Skrypt z zapytaniami SELECT

Po poprawnym wykonaniu poprzednich kroków, można przejść do wprowadzenia zapytań SELECT i przeszukania tabel w celu wydobycia konkretnych danych.

Skrypt **16.SELECT_statement.sql** zawiera 28 zapytań służących użyciu utworzonych wcześniej perspektyw oraz przedstawieniu przykładowych wskaźników biznesowych, istotnych z punktu widzenia danego użytkownika.

Zapytania podzielono na 4 grupy ze względu na 4 użytkowników:

- kursant z identyfikatorem nr 1015,
- pracownik recepcji,
- kierownik ośrodka,
- nadzorca pojazdów.

Pierwszym zapytaniem w każdej z grup jest użycie utworzonej wcześniej perspektywy według formuły:

*SELECT * FROM nazwa_perspektywy*

Aby użyć perspektyw oraz wyświetlić konkretne dane, należy otworzyć skrypt **16.SELECT_statements.sql** oraz uruchomić poszczególne zapytania.

Brak błędów i wyświetlanie wyników takich, jak te zaprezentowane poniżej oznacza, że utworzone perspektywy oraz zdania SELECT są skonstruowane prawidłowo, a sama baza danych została zainstalowana poprawnie.

W celu de-instalacji projektu, po skończonej pracy należy wgrać i uruchomić skrypt **17.OSK_DB_DROP**.

Zapytania SELECT oraz uzyskiwane wyniki (lub ich fragmenty):

```
--perspektywa kursanta id=1015
SELECT *
FROM kursant1015;
```

Twoj identyfikator	Kategoria	Identyfikator kursu	Data rozpoczęcia kursu	Do opłaty	Termin na wpłatę	Status opłaty	Data wpływu
1	1015 CE	10400	23/04/01	2000	23/04/10	ZREALIZOWANA	23/04/04

```
--najbliższe 3 kursy
SELECT ROWNUM, id_kursu, data_roz poczenia, typ_kategorii, cena, liczba_miejsc,
CASE
    WHEN data_roz poczenia > sysdate THEN 'planowany' ELSE 'w trakcie'
END
AS opis
FROM kurs
WHERE data_roz poczenia > sysdate AND ROWNUM <4;
```

ROWNUM	ID_KURSU	DATA_ROZPOCZECIA	TYP_KATEGORII	CENA	LICZBA_MIEJSC	OPIS
1	1	11600 24/04/01	C	2000	20	planowany
2	2	11700 24/05/01	CE	2000	20	planowany
3	3	11800 24/06/01	D	2000	20	planowany

```

--najblizsze kursy i dostepne miejsca
SELECT ROWNUM, kurs.id_kursu, data_roz poczenia, typ_kategorii, cena, liczba_miejsc, COALESCE(lk,0) AS liczba_kursantow,
      liczba_miejsc - COALESCE(lk, 0) AS dostepne_miejsca,
CASE
  WHEN data_roz poczenia > sysdate THEN 'planowany' ELSE 'w trakcie'
END
  AS opis
FROM kurs
FULL OUTER JOIN(SELECT COUNT(relation_1.id_kursanta) AS lk, relation_1.id_kursu
  FROM relation_1
  GROUP BY id_kursu) nazwa ON kurs.id_kursu = nazwa.id_kursu
WHERE data_roz poczenia > sysdate;

```

ROWNUM	ID_KURSU	DATA_ROZPOCZECIA	TYP_KATEGORII	CENA	LICZBA_MIEJSC	LICZBA_KURSANTOW	DOSTEPNE_MIEJSCA	OPIS
1	1	11600 24/04/01	C	2000	20	0	20	planowany
2	2	11700 24/05/01	CE	2000	20	0	20	planowany
3	3	11800 24/06/01	D	2000	20	0	20	planowany
4	4	11900 24/07/01	T	2000	20	0	20	planowany
5	5	12000 24/08/01	B	2000	20	0	20	planowany

```

--perspektywa pracownika recepcji
SELECT *
FROM pracownik_recepcji;

```

	ID_KURSANTA	STATUS_OPLATY	TYP_KATEGORII	ID_KURSU	TYP_ZAPISU	ADRES_EMAIL
1	1000	ZREALIZOWANA	B	10100	na miejscu	(null)
2	1001	ZREALIZOWANA	B	10100	online	edward.b@adres.pl
3	1002	ZREALIZOWANA	B	10100	na miejscu	(null)
4	1003	ZREALIZOWANA	B	10100	online	michal.d@adres.pl
5	1004	ZREALIZOWANA	B	10100	na miejscu	(null)
6	1005	ZREALIZOWANA	B	10100	online	ewa.f@adres.pl
7	1006	ZREALIZOWANA	A	10200	na miejscu	(null)
8	1007	ZREALIZOWANA	A	10200	online	joanna.h@adres.pl
9	1008	ZREALIZOWANA	A	10200	na miejscu	(null)
10	1009	ZREALIZOWANA	A	10200	online	wiktoria.k@adres.pl
11	1010	ZREALIZOWANA	A	10200	na miejscu	(null)
12	1011	ZREALIZOWANA	A	10200	online	krzysztof.m@adres.pl
13	1012	ZREALIZOWANA	C	10300	na miejscu	(null)
14	1013	ZREALIZOWANA	C	10300	online	lukasz.o@adres.pl
15	1014	ZREALIZOWANA	C	10300	na miejscu	(null)
16	1015	ZREALIZOWANA	CE	10400	online	klaudia.r@adres.pl
17	1016	ZREALIZOWANA	CE	10400	na miejscu	(null)
18	1017	ZREALIZOWANA	CE	10400	online	zofia.t@adres.pl
19	1018	ZREALIZOWANA	D	10500	na miejscu	(null)
20	1019	ZREALIZOWANA	D	10500	online	marta.z@adres.pl


```
--wszystkie kursy i dostepne miejsca
SELECT kurs.id_kursu, data_roz poczenia, typ_kategorii, cena, liczba_miejsc, COALESCE(1k,0) AS liczba_kursantow,
       liczba_miejsc - COALESCE(1k, 0) AS dostepne_miejsca,
CASE
    WHEN data_roz poczenia > sysdate THEN 'planowany' ELSE 'w trakcie'
END
AS opis
FROM kurs
FULL OUTER JOIN(SELECT COUNT(relation_1.id_kursanta) AS 1k, relation_1.id_kursu
FROM relation_1
GROUP BY id_kursu) nazwa ON kurs.id_kursu = nazwa.id_kursu;
```

	ID_KURSU	DATA_ROZPOCZECIA	TYP_KATEGORII	CENA	LICZBA_MIEJSC	LICZBA_KURSANTOW	DOSTEPNE_MIEJSCA	OPIS
1	10100	23/01/01	B	2000	20	6	14	w trakcie
2	10200	23/02/01	A	2000	20	6	14	w trakcie
3	10300	23/03/01	C	2000	20	3	17	w trakcie
4	10400	23/04/01	CE	2000	20	3	17	w trakcie
5	10500	23/05/01	D	2000	20	3	17	w trakcie
6	10600	23/06/01	T	2000	20	3	17	w trakcie
7	10700	23/07/01	B	2000	20	3	17	w trakcie
8	10800	23/08/01	B	2000	20	3	17	w trakcie
9	10900	23/09/01	A	2000	20	3	17	w trakcie
10	11000	23/10/01	CE	2000	20	3	17	w trakcie
11	11100	23/11/01	CE	2000	20	3	17	w trakcie
12	11200	23/12/01	B	2000	20	3	17	w trakcie
13	11300	24/01/01	B	2000	20	3	17	w trakcie
14	11400	24/02/01	B	2000	20	6	14	w trakcie
15	11500	24/03/01	A	2000	20	0	20	w trakcie
16	11600	24/04/01	C	2000	20	0	20	planowany
17	11700	24/05/01	CE	2000	20	0	20	planowany
18	11800	24/06/01	D	2000	20	0	20	planowany
19	11900	24/07/01	T	2000	20	0	20	planowany
20	12000	24/08/01	B	2000	20	0	20	planowany

```
-- liczba wszystkich zapisow online i offline
SELECT
liczba_zapisow_online,
liczba_zapisow_offline
FROM
(SELECT COUNT(*) AS liczba_zapisow_online
FROM (SELECT imie, nazwisko, typ_zapisu
FROM kursant, relation_1 WHERE typ_zapisu = 'online' AND kursant.id_kursanta = relation_1.id_kursanta
)
),
(SELECT COUNT(*) AS liczba_zapisow_offline
FROM (SELECT imie, nazwisko, typ_zapisu
FROM kursant, relation_1 WHERE typ_zapisu = 'na miejscu' AND kursant.id_kursanta = relation_1.id_kursanta
)
);
```

	❖	LICZBA_ZAPISOW_ONLINE	❖	LICZBA_ZAPISOW_OFFLINE
1		25		26

```
--liczba instruktorow z uprawnieniami do danej kategorii
SELECT typ_kategorii, COUNT(id_instruktora) AS liczba_instruktorow_z_uprawnieniami
FROM relation_2
GROUP BY typ_kategorii;
```

	TYP_KATEGORII	LICZBA_INSTRUKTOROW_Z_UPRAWNIENIAMI
1	B	10
2	C	5
3	CE	3
4	A	4
5	D	2
6	T	3

```
--perspektywa kierownika
```

```
SELECT *
FROM kierownik_osrodka;
```

ID_KURSANTA	ID_KURSU	TYP_KATEGORII	OCENA_W_SKALI_IDOS	POZYTYWY	NEGATYWY	SUGESTIE
1	1000	10100 B		4 dobry instruktor	wczesna godzina zajesc	pozniejsze godziny zajec
2	1001	10100 B		5 latwy kontakt z biurem	(null)	brak
3	1002	10100 B		4 (null)	(null)	(null)
4	1003	10100 B		4 dobry instruktor	(null)	(null)
5	1004	10100 B		3 (null)	trudny kontakt z insturktorem	(null)
6	1005	10100 B		5 latwy kontakt z biurem	(null)	(null)
7	1006	10200 A		5 solidne przygotowanie do egzaminu	(null)	dobrze by był,o wykonac telefon z przypomnieniem o dokonaniu wpłaty
8	1007	10200 A		4 (null)	trudnosc w umowieniu godzin jazdy	(null)
9	1008	10200 A		5 (null)	(null)	(null)
10	1009	10200 A		5 latwy kontakt z biurem	(null)	pozniejsze godziny zajec
11	1010	10200 A		4 (null)	wykłady rozciagniete w czasie	(null)
12	1011	10200 A		5 dobry instruktor	(null)	(null)
13	1012	10300 C		4 (null)	(null)	(null)
14	1013	10300 C		4 solidne przygotowanie do egzaminu	(null)	dobrze by był,o wykonac telefon z przypomnieniem o dokonaniu wpłaty
15	1014	10300 C		3 (null)	wykłady rozciagniete w czasie	(null)
16	1015	10400 CE		5 latwy kontakt z biurem	(null)	(null)
17	1016	10400 CE		5 (null)	(null)	brak
18	1017	10400 CE		4 dobry instruktor	(null)	(null)
19	1018	10500 D		5 dobry instruktor	(null)	(null)
20	1019	10500 D		5 (null)	(null)	pozniejsze godziny zajec

```
-- liczba kursantow ktorzy zdali za pierwszym razem wszystkie egzaminy
SELECT COUNT(*) AS liczba_kursantow_ktorzy_zdali_wszystkie_egzaminy_za_pierwszym_razem
FROM (
    SELECT id_kursanta, COUNT(id_egzaminu) AS liczba_egzaminow
    FROM egzamin
    GROUP BY id_kursanta
    HAVING COUNT(id_egzaminu) = '4');
```

	LICZBA_KURSANTOW_KTORZY_ZDALI_WSZYSTKIE_EGZAMINY_ZA_PIERWSZYM_RAZEM
1	11

```
-- statystyki zwiazane z kursantami
SELECT liczba_wszystkich_kursantow, liczba_kursantow_ktorzy_zdali_praktyczny_egzamin_panstwowy,
       liczba_kurantow_ktorzy_wypeelnili_ankiete
FROM
  (SELECT COUNT (id_kursanta) AS liczba_wszystkich_kursantow
   FROM kursant),
  (SELECT COUNT(id_kursanta) AS liczba_kursantow_ktorzy_zdali_praktyczny_egzamin_panstwowy
   FROM egzamin
   WHERE typ = 'ZEWNETRZNY' AND rodzaj = 'PRAKTYKA' AND wynik = 'ZDANY'),
  (SELECT COUNT (id_kursanta) AS liczba_kurantow_ktorzy_wypeelnili_ankiete
   FROM ankieta);
```

	LICZBA_WSZYSTKICH_KURSANTOW	LICZBA_KURSANTOW_KTORZY_ZDALI_PRAKTYCZNY_EGZAMIN_PANSTWOWY	LICZBA_KURANTOW_KTORZY_WYPELNILI_ANKIETE
1	51	42	39

```
--liczba oplat z danym statusem
SELECT status, COUNT(id_oplaty) AS liczba
FROM oplata
GROUP BY status;
```

	STATUS	LICZBA
1	ZREALIZOWANA	51

```
--liczba oplat zrealizowanych
SELECT COUNT(id_oplaty) as liczba_oplat_zrealizowanych
FROM oplata
WHERE status = 'ZREALIZOWANA';
```

	LICZBA_OPLAT_ZREALIZOWANYCH
1	51

```
--liczba oplat niezrealizowanych
SELECT COUNT(id_oplaty) AS oplaty_niezrealizowane
FROM oplata
WHERE status <> 'ZREALIZOWANA';
```

	OPLATY_NIEZREALIZOWANE
1	0

```
--statystyki dochodow w poszczegolnych miesiacach
SELECT rok1 AS rok, miesiac1 AS miesiac, COALESCE(liczba_wplat,0) AS "liczba wplat", COALESCE(suma_wplat,0) AS "suma wplat",
       COALESCE(liczba_napraw,0) AS "liczba napraw", COALESCE(suma_kosztow,0) AS "suma kosztow",
       COALESCE(suma_wplat,0) - COALESCE(suma_kosztow,0) AS "bilans w miesiacu"
FROM
  (SELECT
    EXTRACT(YEAR FROM data_wplywu)AS rok1,
    EXTRACT(MONTH FROM data_wplywu)AS miesiac1,
    COUNT(id_oplaty) AS liczba_wplat,
    SUM(kwota) AS suma_wplat
  FROM oplata
  GROUP BY
    EXTRACT(YEAR FROM data_wplywu),
    EXTRACT(MONTH FROM data_wplywu)
  )
FULL OUTER JOIN
  (SELECT
    EXTRACT(YEAR FROM data_rozpczecie)AS rok2,
    EXTRACT(MONTH FROM data_rozpczecie)AS miesiac2,
    COUNT(id_naprawy) AS liczba_napraw,
    SUM(koszt) AS suma_kosztow
  FROM naprawa
  GROUP BY
    EXTRACT(YEAR FROM data_rozpczecie),
    EXTRACT(MONTH FROM data_rozpczecie)
  ) ON rok1 = rok2 AND miesiac1 = miesiac2
ORDER BY rok ASC, miesiac ASC;
```

	ROK	MIESIAC	liczba wplat	suma wplat	liczba_napraw	suma_kosztow	bilans w miesiacu
1	2023	1	6	12000	0	0	12000
2	2023	2	6	12000	3	900	11100
3	2023	3	3	6000	0	0	6000
4	2023	4	3	6000	0	0	6000
5	2023	5	3	6000	0	0	6000
6	2023	6	3	6000	0	0	6000
7	2023	7	3	6000	0	0	6000
8	2023	8	3	6000	0	0	6000
9	2023	9	3	6000	0	0	6000
10	2023	10	3	6000	12	5850	150
11	2023	11	3	6000	0	0	6000
12	2023	12	3	6000	6	4200	1800
13	2024	1	3	6000	17	11550	-5550
14	2024	2	6	12000	15	10000	2000
15	(null)	(null)	0	0	7	1790	-1790

```
--dochod ogolem
SELECT
wplywy_ogolem,
kwota_wszystkich_napraw,
wplywy_ogolem - kwota_wszystkich_napraw AS zysk
FROM
  (SELECT SUM(kwota) AS wplywy_ogolem FROM oplata),
  (SELECT SUM(koszt) AS kwota_wszystkich_napraw FROM naprawa);
```

	WPLYWY_OGOLEM	KWOTA_WSZYSTKICH_NAPRAW	ZYSK
1	102000	34290	67710

```
--statystyki z ankiet
SELECT
liczba_ankiet,
srednia_ocen,
liczba_sugestii,
liczba_pozytywnych_uwag,
liczba_negatywnych_uwag
FROM (SELECT COUNT(id_ankiety) AS liczba_ankiet, round(AVG(ocena_w_skali_1do5),1) AS srednia_ocen
      FROM ankiet),
      (SELECT COUNT(sugestia) AS liczba_sugestii
      FROM ankiet WHERE sugestia IS NOT NULL),
      (SELECT COUNT(pozytyw) AS liczba_pozytywnych_uwag
      FROM ankiet WHERE pozytyw IS NOT NULL),
      (SELECT COUNT(negatyw) AS liczba_negatywnych_uwag
      FROM ankiet WHERE negatyw IS NOT NULL);
```

	⚡ LICZBA_ANKIET	⚡ SREDNIA_OCEN	⚡ LICZBA_SUGESTII	⚡ LICZBA_POZYTYWNYCH_UWAG	⚡ LICZBA_NEGATYWNYCH_UWAG
1	39	4,4	13	19	11

```
--wszystkie niezdane praktyczne egzaminy panstwowe
SELECT id_kursanta, id_egzaminu, data_egzaminu, typ, rodzaj, wynik, powod_niezaliczenia
FROM egzamin
WHERE typ = 'ZEWNETRZNY' AND rodzaj = 'PRAKTYKA' AND powod_niezaliczenia IS NOT NULL;
```

	⚡ ID_KURSANTA	⚡ ID_EGZAMINU	⚡ DATA_EGZAMINU	⚡ TYP	⚡ RODZAJ	⚡ WYNIK	⚡ POWOD_NIEZALICZENIA
1	1002	40112	23/03/22	ZEWNETRZNY	PRAKTYKA	NIEZDANY	nieudane parkowanie
2	1003	40118	23/03/29	ZEWNETRZNY	PRAKTYKA	NIEZDANY	najechanie na tyczke
3	1006	40132	23/04/22	ZEWNETRZNY	PRAKTYKA	NIEZDANY	nieudane parkowanie
4	1007	40138	23/04/29	ZEWNETRZNY	PRAKTYKA	NIEZDANY	najechanie na tyczke
5	1010	40152	23/04/22	ZEWNETRZNY	PRAKTYKA	NIEZDANY	najechanie na tyczke
6	1011	40158	23/04/29	ZEWNETRZNY	PRAKTYKA	NIEZDANY	najechanie na tyczke
7	1014	40172	23/05/31	ZEWNETRZNY	PRAKTYKA	NIEZDANY	nieudane parkowanie
8	1015	40178	23/06/11	ZEWNETRZNY	PRAKTYKA	NIEZDANY	najechanie na tyczke
9	1018	40192	23/07/10	ZEWNETRZNY	PRAKTYKA	NIEZDANY	nieudane parkowanie
10	1019	40198	23/07/11	ZEWNETRZNY	PRAKTYKA	NIEZDANY	najechanie na tyczke

```
--powody niezaliczenia praktycznych egzaminow panstwowowych oraz liczba ich wystapien
SELECT powod_niezaliczenia AS powod_niezaliczenia_egzaminu_panstwowego, COUNT(*) AS liczba_wystapien
FROM egzamin
WHERE typ = 'ZEWNETRZNY' AND powod_niezaliczenia IS NOT NULL
GROUP BY powod_niezaliczenia;
```

⚡ POWOD_NIEZALICZENIA_EGZAMINU_PANSTWOWEGO	⚡ LICZBA_WYSTAPIEN
1 najechanie na tyczke	13
2 nieudane parkowanie	7

```
--perspektywa nadzorcy pojazdow
SELECT *
FROM nadzorca_pojazdow;
```

ID_POJAZDU	MARKA	MODEL	TYP_KATEGORII	STATUS_NAPRAWY	Data rozpoczęcia naprawy	STATUS_PRZEGLADU	TERMIN_NASTEPNEGO_PRZEGLADU
1	413 Suzuki	SFV 650UA 35kW	A2	W TRAKCIE	24/01/26	NIEAKTUALNY	24/01/18
2	408 Hyundai	i20	B	DO ODBIORU	23/02/27	NIEAKTUALNY	24/01/22
3	404 Hyundai	i20	B	W TRAKCIE	23/02/28	NIEAKTUALNY	24/01/25
4	421 MAN	TGL 12	CE	W TRAKCIE	24/02/26	NIEAKTUALNY	24/02/05
5	406 Hyundai	i20	B	DO ODBIORU	24/02/28	AKTUALNY	24/04/08
6	414 Honda	CBF 125M	A1	W TRAKCIE	24/01/21	AKTUALNY	24/04/08
7	425 ZETOR	PROXIMA PLUS 80	T	DO ODBIORU	24/03/01	AKTUALNY	24/04/08
8	412 Suzuki	SFV 650UA 35kW	A2	DO ODBIORU	24/01/16	AKTUALNY	24/04/08
9	422 DAF	LF	CE	DO ODBIORU	24/01/17	AKTUALNY	24/04/08
10	420 DAF	LF	C	W TRAKCIE	24/03/01	AKTUALNY	24/06/04
11	403 Hyundai	i20	B	W TRAKCIE	24/01/25	AKTUALNY	24/06/04
12	411 Suzuki	SFV 650UA 35kW	A2	W TRAKCIE	24/02/27	AKTUALNY	24/08/14
13	407 Hyundai	i20	B	W TRAKCIE	24/03/04	AKTUALNY	24/08/14
14	424 NEW HOLLAND	T4.65	T	W TRAKCIE	24/03/06	AKTUALNY	24/08/14
15	410 Suzuki	SV 650	A	W TRAKCIE	24/03/02	AKTUALNY	25/01/12
16	423 Solbus	ST 11	D	DO ODBIORU	24/01/27	AKTUALNY	25/01/12
17	401 Hyundai	i20	B	W TRAKCIE	24/02/28	AKTUALNY	25/01/12
18	417 Renault	Trafic	BE	W TRAKCIE	24/03/05	AKTUALNY	25/01/12
19	418 MAN	TGL 12.220	C	W TRAKCIE	23/02/26	AKTUALNY	25/02/04
20	416 Kymco	Agility 50	AM	DO ODBIORU	24/02/29	AKTUALNY	25/02/10
21	402 Hyundai	i20	B	W TRAKCIE	24/03/03	AKTUALNY	25/02/10

```
-- dane pojazdow ktore byly naprawiane wiecej niz 2 razy oraz koszt wszystkich napraw i sredni koszt jednej naprawy danego pojazdu
-- uporządkowane malejaco wg kosztu wszystkich napraw
SELECT pojazd.*, liczba_napraw, koszt_wszystkich_napraw_pojazdu, sredni_koszt_jednej_naprawy_pojazdu
FROM pojazd
INNER JOIN (SELECT id_pojazdu, COUNT(id_naprawy) AS liczba_napraw, SUM(koszt) AS koszt_wszystkich_napraw_pojazdu,
round(AVG(koszt),1) AS sredni_koszt_jednej_naprawy_pojazdu
FROM naprawa
GROUP BY id_pojazdu
HAVING COUNT(id_naprawy) > 2) tabelal ON pojazd.id_pojazdu = tabelal.id_pojazdu
ORDER BY koszt_wszystkich_napraw_pojazdu DESC;
```

ID_P...	TYP	MARKA	MO...	ROK_P...	STATUS_P...	TERMIN_N...	DANE_PRZYPISY	TYP_K...	LICZBA_NAPRAW	KOSZT_WSZYSTKICH_NAPRAW_POJAZDU	SREDNI_KOSZT_JEDNEJ_NAPRAWY_POJAZDU
1	422 Samochod...	DAF	LF	2018	AKTUALNY	24/04/08	przyczepa typu tandem	CE	3	2700	900
2	423 Autobus	Sol...	ST 11	2019	AKTUALNY	25/01/12	(null)	D	4	2650	662,5
3	402 Samochod...	Hyu...	i20	2019	AKTUALNY	25/02/10	(null)	B	3	2100	700
4	425 Ciągnik ...	ZETOR	PRO...	2019	AKTUALNY	24/04/08	przyczepa MAR-POL MT 600 T		3	1950	650
5	404 Samochod...	Hyu...	i20	2021	NIEAKTUALNY	24/01/25	(null)	B	3	1900	633,3
6	424 Ciągnik ...	NEW...	T4.65	2020	AKTUALNY	24/08/14	przyczepa PRONAR T663/2	T	3	1680	560
7	406 Samochod...	Hyu...	i20	2020	AKTUALNY	24/04/08	(null)	B	3	1650	550
8	405 Samochod...	Hyu...	i20	2022	AKTUALNY	24/10/12	(null)	B	3	1550	516,7
9	403 Samochod...	Hyu...	i20	2019	AKTUALNY	24/06/04	(null)	B	3	1350	450


```
--liczba pojazdow o danym statusie przegladu
SELECT status_przegladu, COUNT(*) AS liczba_pojazdow
FROM pojazd
GROUP BY status_przegladu;
```

	STATUS_PRZEGladU	LICZBA_POJAZDOW
1	AKTUALNY	21
2	NIEAKTUALNY	4

```
--pojazdy do oddania na przeglad
SELECT id_pojazdu AS pojazdy_do_oddania_na_przeglad, status_przegladu, termin_nastepnego_przegladu
FROM pojazd
WHERE status_przegladu = 'NIEAKTUALNY' OR (termin_nastepnego_przegladu >= sysdate AND termin_nastepnego_przegladu <= sysdate + 31)
ORDER BY termin_nastepnego_przegladu;
```

	POJAZDY_DO_ODDANIA_NA_PRZEGlad	STATUS_PRZEGladU	TERMIN_NASTEPNEGO_PRZEGladU
1	413	NIEAKTUALNY	24/01/18
2	408	NIEAKTUALNY	24/01/22
3	404	NIEAKTUALNY	24/01/25
4	421	NIEAKTUALNY	24/02/05
5	406	AKTUALNY	24/04/08
6	412	AKTUALNY	24/04/08
7	425	AKTUALNY	24/04/08
8	414	AKTUALNY	24/04/08
9	422	AKTUALNY	24/04/08

```
--pojazdy z nieaktualnym przegladem
SELECT id_pojazdu AS pojazdy_z_nieaktualnym_przegladem, status_przegladu, termin_nastepnego_przegladu
FROM pojazd
WHERE status_przegladu = 'NIEAKTUALNY'
ORDER BY termin_nastepnego_przegladu;
```

	POJAZDY_Z_NIEAKTUALNYM_PRZEGladEM	STATUS_PRZEGladU	TERMIN_NASTEPNEGO_PRZEGladU
1	413	NIEAKTUALNY	24/01/18
2	408	NIEAKTUALNY	24/01/22
3	404	NIEAKTUALNY	24/01/25
4	421	NIEAKTUALNY	24/02/05

```
-- pojazdy z bliskim terminem przeglądu, tj. w najbliższym miesiącu
SELECT id_pojazdu AS pojazdy_ze_zblizajacym_sie_terminem_przeglądu, status_przeglądu, termin_nastepnego_przeglądu
FROM pojazd
WHERE termin_nastepnego_przeglądu >= sysdate AND termin_nastepnego_przeglądu <= sysdate + 31
ORDER BY termin_nastepnego_przeglądu;
```

	POJAZDY_ZE_ZBLIZAJACYM_SIE_TERMINEM_PRZEGŁADU	STATUS_PRZEGŁADU	TERMIN_NASTĘPNEGO_PRZEGŁADU
1	406	AKTUALNY	24/04/08
2	412	AKTUALNY	24/04/08
3	425	AKTUALNY	24/04/08
4	422	AKTUALNY	24/04/08
5	414	AKTUALNY	24/04/08

```
--pojazdy gotowe do odbioru z naprawy
SELECT id_pojazdu AS id_pojazdow_gotowych_do_odbioru_z_naprawy
FROM naprawa
WHERE status = 'DO ODBIORU'
GROUP BY id_pojazdu;
```

	ID_POJAZDOW_GOTOWYCH_DO_ODBIORU_Z_NAPRAWY
1	412
2	422
3	408
4	425
5	423
6	406
7	416

```
-- 3 najczesciej wystepujace naprawy
SELECT ROWNUM, opis_naprawy, ilosc_wystapien_naprawy
FROM (
SELECT opis AS opis_naprawy, COUNT(*) AS ilosc_wystapien_naprawy
FROM naprawa
GROUP BY opis
ORDER BY ilosc_wystapien_naprawy DESC
)
WHERE ROWNUM <4;
```

ROWNUM	OPIS_NAPRAWY	ILOSC_WYSTAPIEN_NAPRAWY
1	1 Wymiana opon	8
2	2 Wymiana swiec zaplonowych	7
3	3 Wymiana oleju i filtrow	7


```

--statystyki napraw
SELECT
liczba_wszystkich_napraw,
koszt_wszystkich_napraw_ogolem,
sredni_koszt_jednej_naprawy,
sredni_koszt_wszystkich_napraw_jednego_pojazdu
FROM
    (SELECT COUNT(id_naprawy) AS liczba_wszystkich_napraw, SUM(koszt) AS koszt_wszystkich_napraw_ogolem,
        AVG(koszt) AS sredni_koszt_jednej_naprawy
        FROM naprawa),

    (SELECT AVG(suma_kosztow_na_pojazd) AS sredni_koszt_wszystkich_napraw_jednego_pojazdu
        FROM (SELECT id_pojazdu, SUM(koszt) AS suma_kosztow_na_pojazd
            FROM naprawa
            GROUP BY id_pojazdu
        )
    );

```

	⚡ LICZBA_WSZYSTKICH_NAPRAW	⚡ KOSZT_WSZYSTKICH_NAPRAW_OGOLEM	⚡ SREDNI_KOSZT_JEDNEJ_NAPRAWY	⚡ SREDNI_KOSZT_WSZYSTKICH_NAPRAW_JEDNEGO_POJAZDU
1	60	34290	571,5	1371,6

```

--statystyki napraw dla kazdego pojazdu
SELECT id_pojazdu, COUNT(*) AS liczba_napraw, SUM(koszt) AS suma_kosztow_na_pojazd,
    round(AVG(koszt), 1) AS sredni_koszt_jednej_naprawy_na_pojazd
FROM naprawa
GROUP BY id_pojazdu
ORDER BY suma_kosztow_na_pojazd DESC;

```

	⚡ ID_POJAZDU	⚡ LICZBA_NAPRAW	⚡ SUMA_KOSZTOW_NA_POJAZD	⚡ SREDNI_KOSZT_JEDNEJ_NAPRAWY_NA_POJAZD
1	422	3	2700	900
2	423	4	2650	662,5
3	402	3	2100	700
4	425	3	1950	650
5	404	3	1900	633,3
6	412	2	1800	900
7	424	3	1680	560
8	406	3	1650	550
9	414	2	1600	800
10	405	3	1550	516,7
11	418	2	1500	750
12	408	2	1500	750
13	403	3	1350	450
14	415	2	1300	650
15	416	2	1150	575
16	413	2	1150	575
17	409	2	1100	550
18	419	2	1100	550
19	421	2	900	450
20	411	2	900	450
21	401	2	900	450
22	420	2	600	300
23	410	2	600	300
24	407	2	330	165
25	417	2	330	165