



Politechnika Krakowska im. Tadeusza Kościuszki
Wydział informatyki i telekomunikacji
Katedra informatyki

Stacjonarna wypożyczalnia samochodów

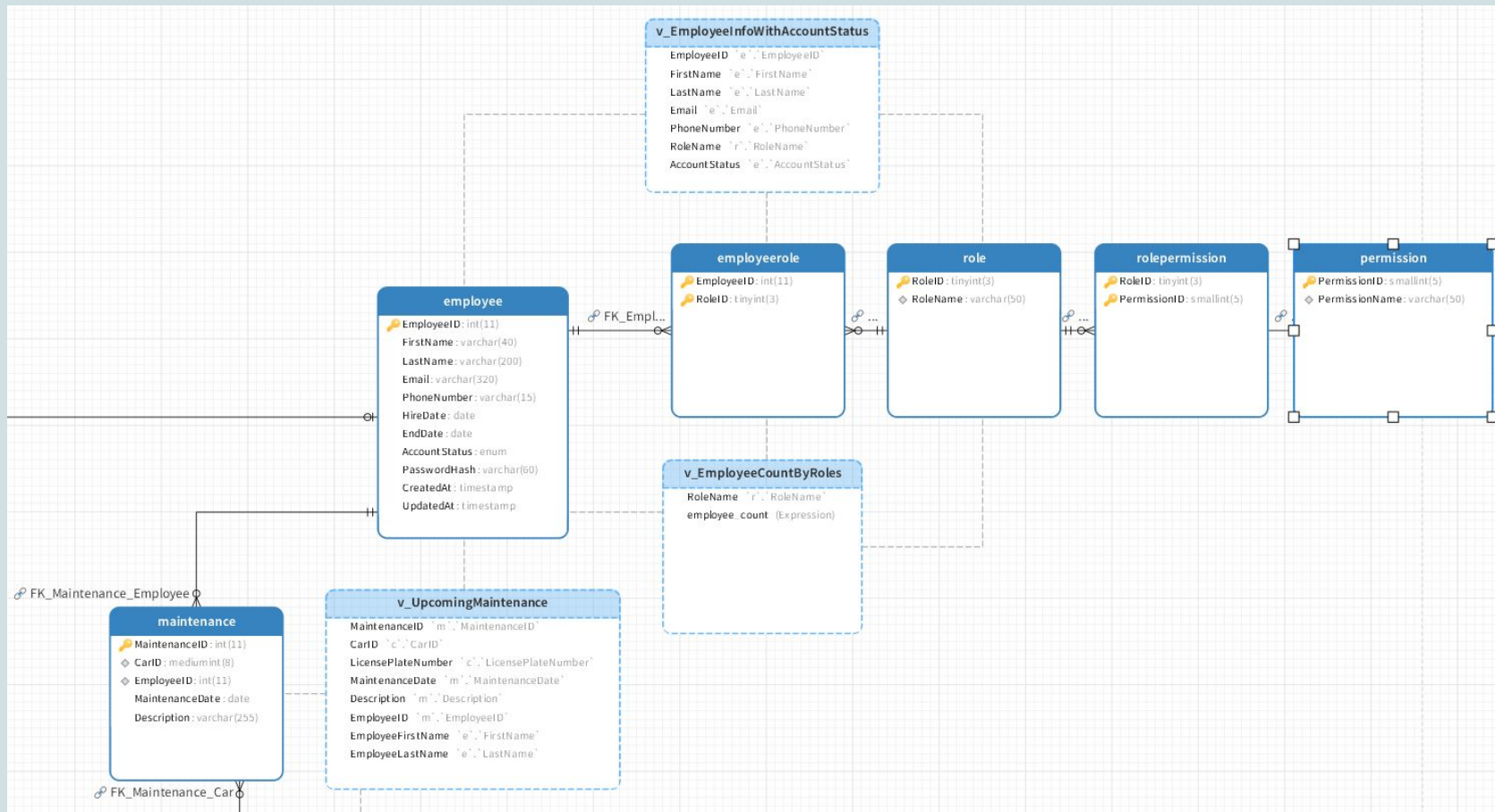
Dyląg Matylda
Herczyk Wiktoria
Huza Martyna

Kraków, 2024

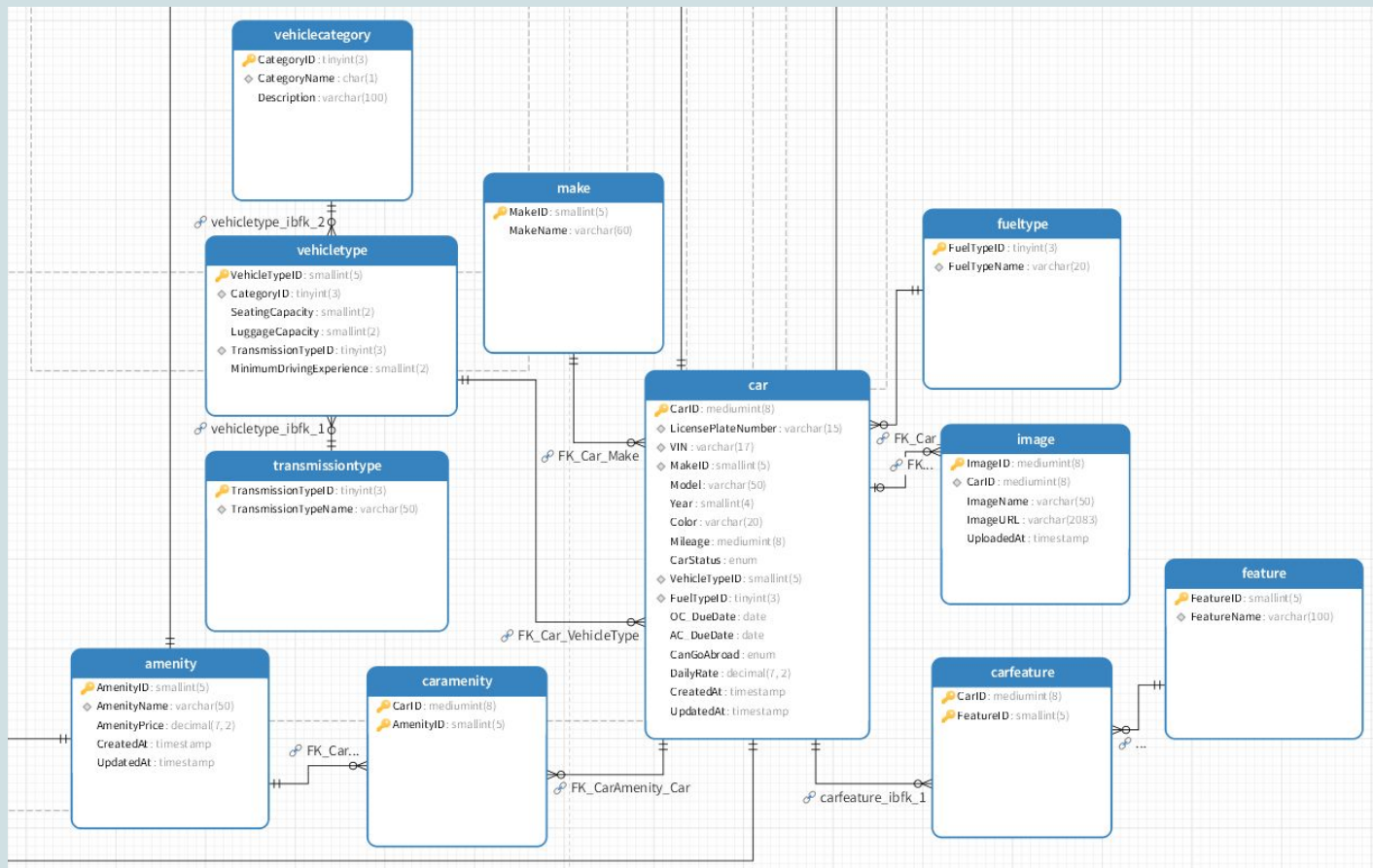
Funkcje bazy

- Zarządzanie wypożyczaniem samochodów (rezerwacja, wynajem)
- Monitorowanie stanu aut (przeglądy)
- Generowanie faktur
- Zarządzanie historią wypożyczeń
- Rejestr dodatkowych udogodnień i dodatków dla poszczególnych samochodów
- Przechowywanie informacji o klientach, samochodach i pracownikach
- Dynamiczne dodawanie ról pracowników

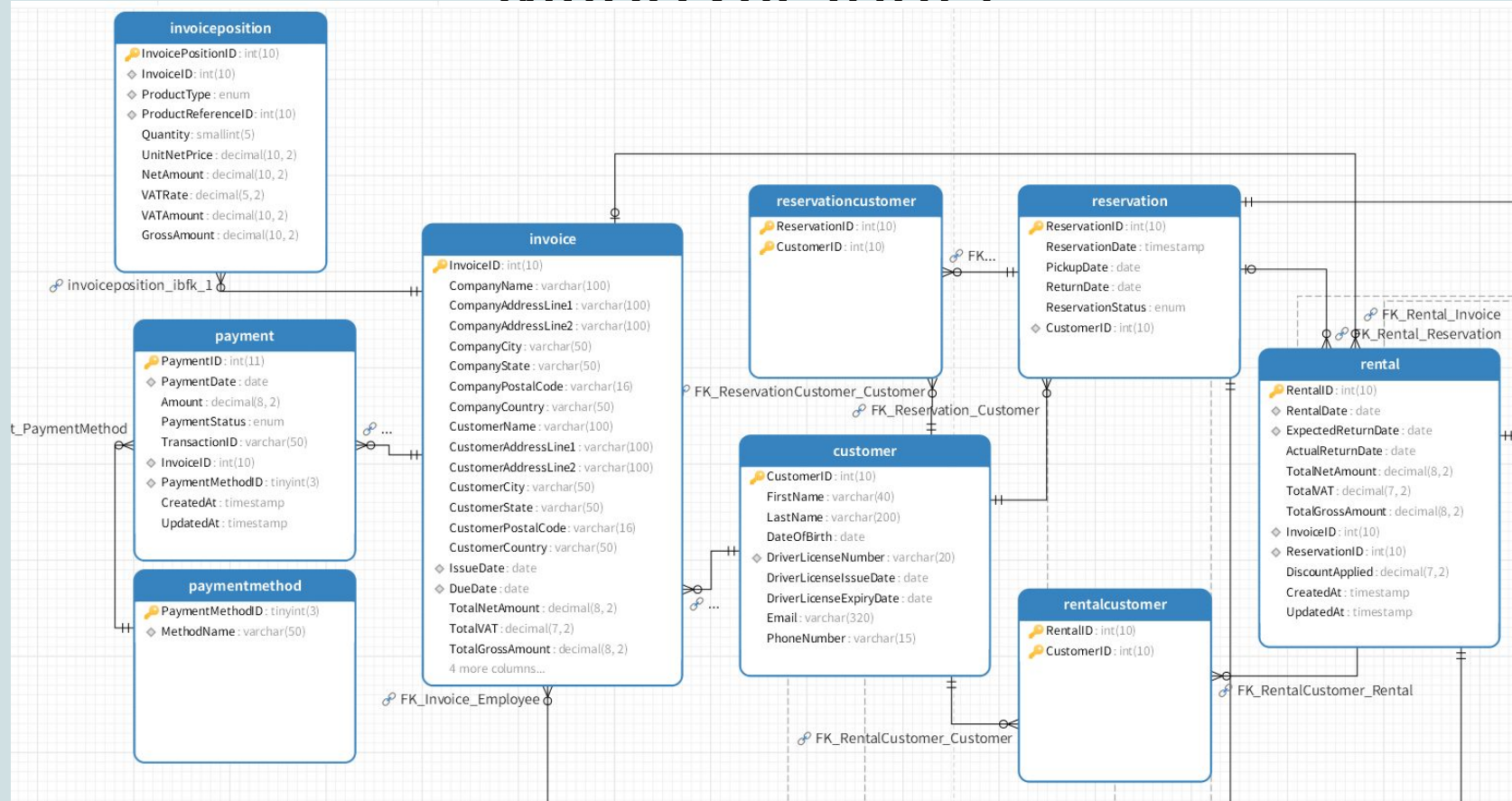
ERD - pracownicy, role, przeglądy



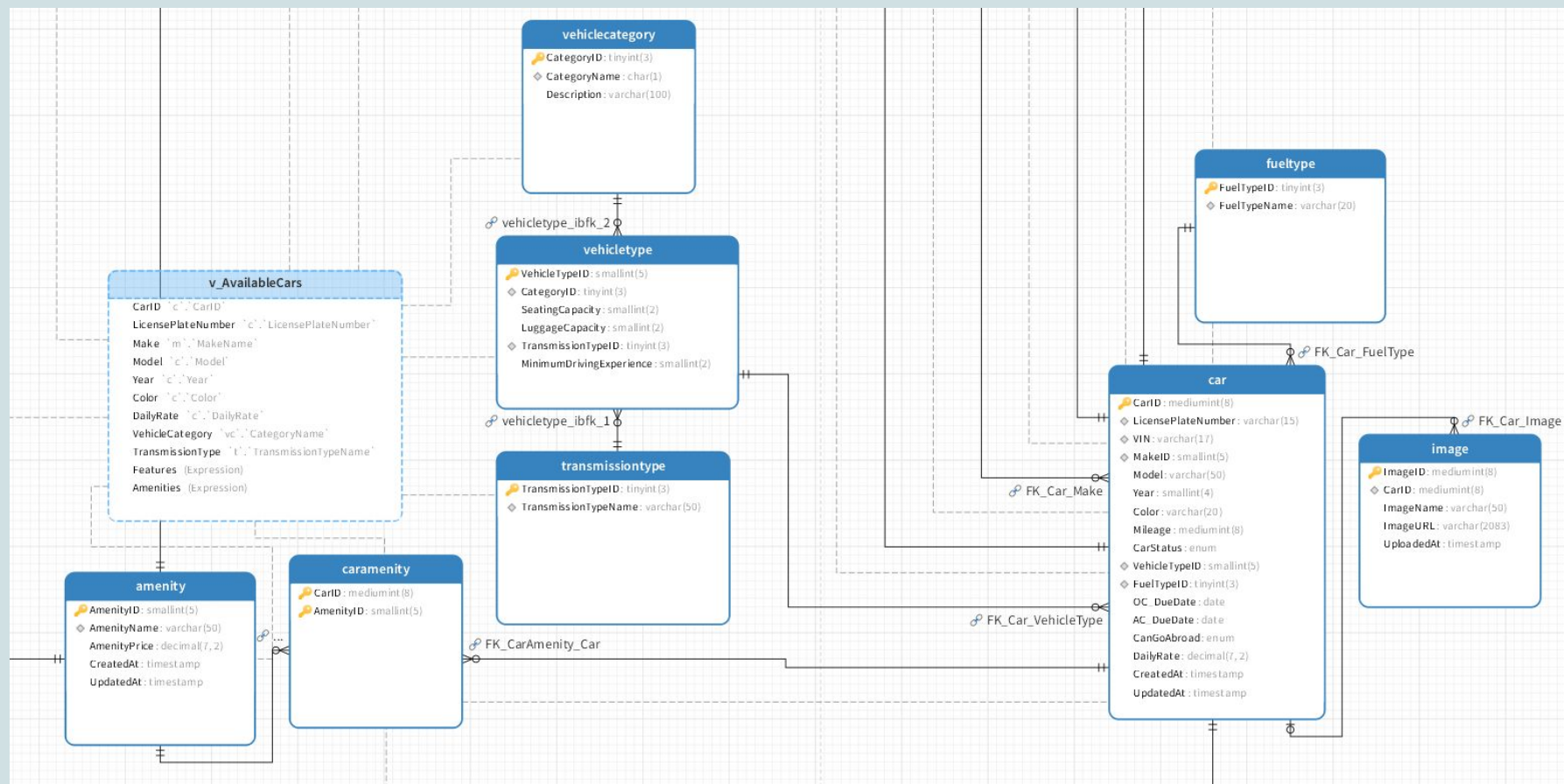
ERD - samochody i udogodnienia



ERD - faktura, klienci, rezerwacje, wynajem



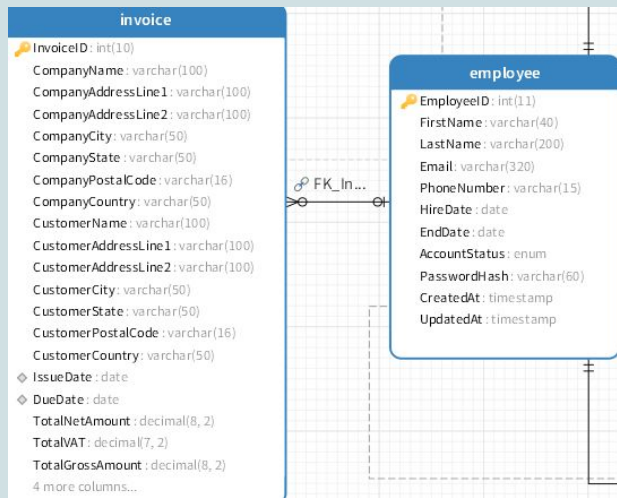
ERD - samochody i udogodnienia



Relacje

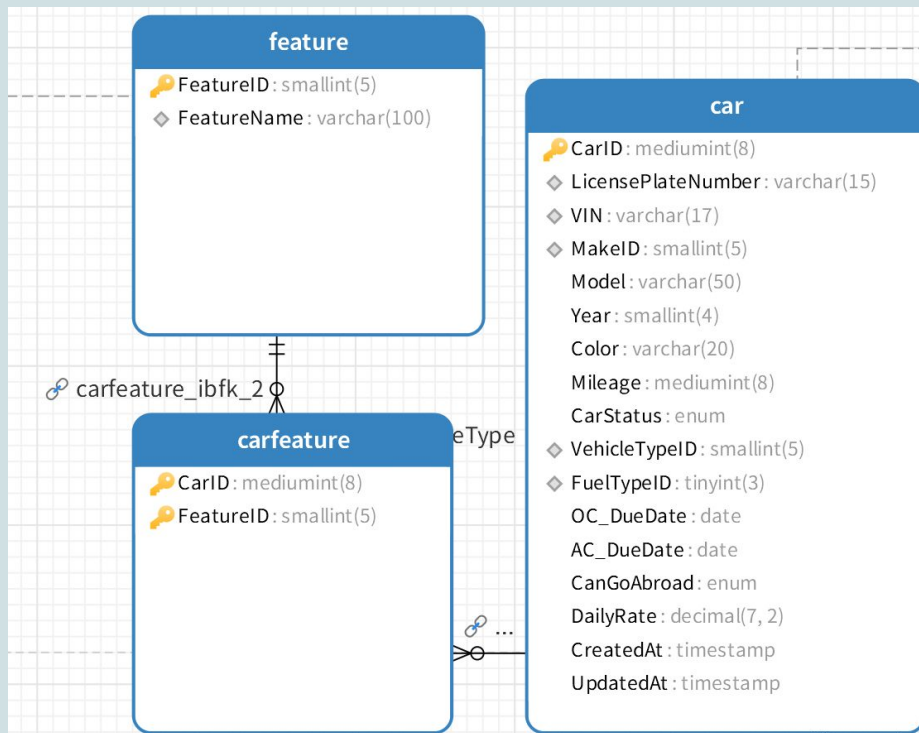


many to many



zero or one to many

Relacje



one and only one
to
zero or many

Najciekawsze widoki

Pracownicy z statusem konta

```
CREATE VIEW AS
SELECT
  `e`.`EmployeeID` AS `EmployeeID`,
  `e`.`FirstName` AS `FirstName`,
  `e`.`LastName` AS `LastName`,
  `e`.`Email` AS `Email`,
  `e`.`PhoneNumber` AS `PhoneNumber`,
  `r`.`RoleName` AS `RoleName`,
  `e`.`AccountStatus` AS `AccountStatus`
FROM ((
  `employee` `e`
  JOIN `employee_role` `er`
    ON (`e`.`EmployeeID` = `er`.`EmployeeID`))
  JOIN `role` `r`
    ON (`r`.`RoleID` = `er`.`RoleID`))
ORDER BY
  `e`.`AccountStatus`
```

EmployeeID	FirstName	LastName	Email	PhoneNumber	RoleName	AccountStatus
12	Dorota	Mazur	dorota.mazur@example.co	512 345 678	Koordynator ds. wynajmu	Aktywne
4	Katarzyna	Wójcik	katarzyna.wojcik@exam	504 567 890	Pracownik biura wynajmu	Aktywne
8	Agnieszka	Krawczyk	agnieszka.krawczyk@exa	508 901 234	Koordynator ds. wynajmu	Aktywne
3	Tomasz	Wiśniewski	tomasz.wisniewski@exam	503 456 789	Pracownik biura wynajmu	Aktywne
10	Paulina	Wojciechowska	paulina.wojciechowska@	510 123 456	Specjalista ds. marketingu	Aktywne
7	Michał	Szymański	michal.szymanski@exam	507 890 123	Koordynator ds. wynajmu	Aktywne
13	Łukasz	Zajac	lukasz.zajac@example.co	513 456 789	Mechanik	Aktywne
9	Krzysztof	Dąbrowski	krzysztof.dabrowski@exa	509 012 345	Specjalista ds. marketingu	Aktywne
1	Marek	Kowalczyk	marek.kowalczyk@exam	501 234 567	Dyrektor Generalny CEO	Aktywne
11	Jacek	Pawlak	jacek.pawlak@example.cc	511 234 567	Mechanik	Aktywne
14	Ewa	Pietrzak	ewa.pietrzak@example.cc	514 567 890	Pracownik biura wynajmu	Aktywne
15	Bartek	Górski	bartek.gorski@example.co	515 678 901	Księgowy	Aktywne
6	Małgorzata	Zielińska	malgorzata.zielinska@ex	506 789 012	Pracownik biura wynajmu	Aktywne
2	Anna	Nowak	anna.nowak@example.co	502 345 678	Księgowy	Aktywne
5	Piotr	Lewandowski	piotr.lewandowski@exam	505 678 901	Pracownik biura wynajmu	Aktywne

Najciekawsze widoki

Miesięczne zarobki

Year	Month	TotalRevenue
2024	11	926,90
2024	9	553,50
2024	1	2029,50

```
CREATE VIEW `v_RevenueByMonth` AS
SELECT
  YEAR (`r`.`RentalDate`) AS `Year`,
  MONTH (`r`.`RentalDate`) AS `Month`,
  sum(`r`.`TotalGrossAmount`) AS `TotalRevenue`
FROM
  `rental` `r`
GROUP BY
  YEAR (`r`.`RentalDate`),
  MONTH (`r`.`RentalDate`)
ORDER BY
  YEAR (`r`.`RentalDate`) DESC,
  MONTH (`r`.`RentalDate`) DESC
```

Najciekawsze widoki

Najpopularniejsze auta

```
CREATE VIEW `v_PopularCars` AS
SELECT
  `c`.`CarID` AS `CarID`,
  `c`.`LicensePlateNumber` AS `LicensePlateNumber`,
  `m`.`MakeName` AS `Make`,
  `c`.`Model` AS `Model`,
  count(`rc`.`RentalID`) AS `TimesRented`
FROM ((
  `car` `c`
  LEFT JOIN `make` `m` ON (`c`.`MakeID` = `m`.`MakeID`))
  LEFT JOIN `rentalcar` `rc` ON (`c`.`CarID` = `rc`.`CarID`))
GROUP BY
  `c`.`CarID`,
  `c`.`LicensePlateNumber`,
  `m`.`MakeName`,
  `c`.`Model`
ORDER BY
  count(`rc`.`RentalID`) DESC
```

CarID	LicensePlate	Make	Model	TimesRented
14	POZ99999	Renault	Clio	1
15	LOD10101	Toyota	Yaris	1
1	KRA12345	Toyota	Corolla	1
8	GDA33333	Honda	Civic	1
9	SZC44444	Hyundai	i30	0
10	KAT55555	Peugot	208	0
11	KRK66666	Nissan	Qashqai	0
12	WAW77777	Kia	Sportage	0
13	GD88888	Mazda	CX-5	0
16	GDA20202	Ford	Focus	0
17	SZC30303	Vauxhall	Astra	0
2	KRK98765	Ford	Fiesta	0
18	KAT40404	BMW	3 Series	0
3	WAW54321	BMW	X5	0
19	KRK50505	Mercedes-	E-Class	0
4	GD12345	Audi	A4	0
20	WAW60606	Audi	Q5	0
5	WRO67890	(Null)	C-Class	0
6	POZ11111	Volkswage	Golf	0
7	LOD22222	Skoda	Octavia	0

Najciekawsze funkcje - sprawdzanie dostępności auta

```
1 CREATE FUNCTION `IsCarAvailable`(input_CarID INT,  
2   input_StartDate DATE,  
3   input_EndDate DATE  
4 ) RETURNS varchar(20) CHARSET utf8mb4 COLLATE utf8mb4_uca1400_ai_ci  
5   DETERMINISTIC  
6 BEGIN  
7   DECLARE rental_conflict INT DEFAULT 0;  
8   DECLARE reservation_conflict INT DEFAULT 0;  
9   DECLARE car_exists INT DEFAULT 0;  
10  
11   SELECT COUNT(*) INTO car_exists FROM car WHERE CarID = input_CarID;  
12  
13   IF car_exists = 0 THEN RETURN 'Car Not Found';  
14   END IF;  
15  
16   SELECT COUNT(*)  
17   INTO rental_conflict  
18   FROM rental r  
19   JOIN rentalcar rc ON r.RentalID = rc.RentalID  
20   WHERE rc.CarID = input_CarID  
21         AND ( (input_StartDate BETWEEN r.RentalDate AND COALESCE(r.ActualReturnDate, r.ExpectedReturnDate)) OR  
22              (input_EndDate BETWEEN r.RentalDate AND COALESCE(r.ActualReturnDate, r.ExpectedReturnDate)) OR  
23              (r.RentalDate BETWEEN input_StartDate AND input_EndDate));  
24  
25   SELECT COUNT(*)  
26   INTO reservation_conflict  
27   FROM reservation res  
28   JOIN reservationcar rc ON res.ReservationID = rc.ReservationID  
29   WHERE rc.CarID = input_CarID  
30         AND res.ReservationStatus = 'Active'  
31         AND ( (input_StartDate BETWEEN res.PickupDate AND res.ReturnDate) OR  
32              (input_EndDate BETWEEN res.PickupDate AND res.ReturnDate) OR  
33              (res.PickupDate BETWEEN input_StartDate AND input_EndDate));  
34  
35   IF rental_conflict > 0 OR reservation_conflict > 0 THEN RETURN 'Not Available';  
36   ELSE RETURN 'Available';  
37   END IF;  
38 END
```

Najciekawsze funkcje - oblicz zniżkę

```
1 CREATE FUNCTION `CalculateDiscountAmount` (RentalStartDate DATE,  
2     ExpectedReturnDate DATE,  
3     TotalNetAmount DECIMAL(8,2)  
4 ) RETURNS decimal(7,2)  
5     DETERMINISTIC  
6 BEGIN  
7     DECLARE RentalDays INT;  
8     DECLARE DiscountAmount DECIMAL(5,2) DEFAULT 0.00;  
9  
10    SET RentalDays = DATEDIFF(ExpectedReturnDate, RentalStartDate) + 1;  
11  
12    SELECT DiscountPercentage INTO DiscountAmount  
13    FROM discount  
14    WHERE RentalDays BETWEEN MinRentalDays AND MaxRentalDays  
15    OR (MaxRentalDays IS NULL AND MinRentalDays <= RentalDays)  
16    ORDER BY MinRentalDays DESC  
17    LIMIT 1;  
18  
19    IF DiscountAmount IS NULL THEN  
20        SET DiscountAmount = 0.00;  
21    END IF;  
22  
23    RETURN ROUND(TotalNetAmount * DiscountAmount / 100, 2);  
24 END
```


Najciekawsze funkcje - generuj fakturę część 1

```
1 CREATE DEFINER='huza_martyna'@`%` PROCEDURE `GenerateInvoiceWithAddresses`(  
2     IN input_RentalID INT UNSIGNED,  
3     IN input_VATRate DECIMAL(5, 2),  
4     IN input_InvoiceStatus VARCHAR(6) CHARACTER SET utf8mb4 COLLATE utf8mb4_uca1400_ai_ci, # Paid or Unpaid but enum  
    is not allowed so varchar it  
5     IN input_CompanyName VARCHAR(100) CHARACTER SET utf8mb4 COLLATE utf8mb4_uca1400_ai_ci,  
6     IN input_CompanyAddressLine1 VARCHAR(100) CHARACTER SET utf8mb4 COLLATE utf8mb4_uca1400_ai_ci,  
7     IN input_CompanyAddressLine2 VARCHAR(100) CHARACTER SET utf8mb4 COLLATE utf8mb4_uca1400_ai_ci,  
8     IN input_CompanyCity VARCHAR(50) CHARACTER SET utf8mb4 COLLATE utf8mb4_uca1400_ai_ci,  
9     IN input_CompanyState VARCHAR(50) CHARACTER SET utf8mb4 COLLATE utf8mb4_uca1400_ai_ci,  
10    IN input_CompanyPostalCode VARCHAR(16) CHARACTER SET utf8mb4 COLLATE utf8mb4_uca1400_ai_ci,  
11    IN input_CompanyCountry VARCHAR(50) CHARACTER SET utf8mb4 COLLATE utf8mb4_uca1400_ai_ci,  
12    IN input_CustomerID INT UNSIGNED,  
13    IN input_CustomerAddressLine1 VARCHAR(100) CHARACTER SET utf8mb4 COLLATE utf8mb4_uca1400_ai_ci,  
14    IN input_CustomerAddressLine2 VARCHAR(100) CHARACTER SET utf8mb4 COLLATE utf8mb4_uca1400_ai_ci,  
15    IN input_CustomerCity VARCHAR(50) CHARACTER SET utf8mb4 COLLATE utf8mb4_uca1400_ai_ci,  
16    IN input_CustomerState VARCHAR(50) CHARACTER SET utf8mb4 COLLATE utf8mb4_uca1400_ai_ci,  
17    IN input_CustomerPostalCode VARCHAR(16) CHARACTER SET utf8mb4 COLLATE utf8mb4_uca1400_ai_ci,  
18    IN input_CustomerCountry VARCHAR(50) CHARACTER SET utf8mb4 COLLATE utf8mb4_uca1400_ai_ci  
19 )  
20 BEGIN  
21     DECLARE new_InvoiceID INT UNSIGNED;  
22     DECLARE invoice_total_net DECIMAL(8, 2) DEFAULT 0.00;  
23     DECLARE invoice_total_vat DECIMAL(7, 2) DEFAULT 0.00;  
24     DECLARE invoice_total_gross DECIMAL(8, 2) DEFAULT 0.00;  
25  
26     # Validate InvoiceStatus input  
27     IF input_InvoiceStatus NOT IN ('Paid', 'Unpaid') THEN  
28         SIGNAL SQLSTATE '45000'  
29         SET MESSAGE_TEXT = 'Invalid InvoiceStatus value. Must be "Paid" or "Unpaid".';  
30     END IF;  
31  
32     SET SESSION TRANSACTION ISOLATION LEVEL REPEATABLE READ;  
33     START TRANSACTION;
```


Najciekawsze funkcje - generuj fakturę część 2

```
35 # Create the invoice
36 INSERT INTO invoice (
37     CompanyName, CompanyAddressLine1, CompanyAddressLine2,
38     CompanyCity, CompanyState, CompanyPostalCode, CompanyCountry,
39     CustomerName, CustomerAddressLine1, CustomerAddressLine2,
40     CustomerCity, CustomerState, CustomerPostalCode, CustomerCountry,
41     CustomerID, IssueDate, DueDate, InvoiceStatus
42 )
43 SELECT
44     input_CompanyName, input_CompanyAddressLine1, input_CompanyAddressLine2,
45     input_CompanyCity, input_CompanyState, input_CompanyPostalCode, input_CompanyCountry,
46     CONCAT(c.FirstName, ' ', c.LastName) AS CustomerName,
47     input_CustomerAddressLine1, input_CustomerAddressLine2,
48     input_CustomerCity, input_CustomerState, input_CustomerPostalCode, input_CustomerCountry,
49     input_CustomerID,
50     CURDATE(),
51     DATE_ADD(CURDATE(), INTERVAL 7 DAY), # Default payment deadline: 7 days
52     input_InvoiceStatus
53 FROM customer c
54 WHERE c.CustomerID = input_CustomerID;
55
56 # Get the newly inserted InvoiceID to use later with the invoicePosition
57 SELECT InvoiceID
58 INTO new_InvoiceID
59 FROM invoice
60 WHERE CustomerID = input_CustomerID
61     AND IssueDate = CURDATE()
62     AND DueDate = DATE_ADD(CURDATE(), INTERVAL 7 DAY)
63     AND InvoiceStatus = input_InvoiceStatus
64     AND InvoiceID = LAST_INSERT_ID(); # This will work. Why? Cuz it's for a given session (no need to worry about
the other users) and since the last insert was into an invoice table (the insert above) we will get the last
invoice ID
```

Najciekawsze funkcje - generuj fakturę część 3

```
67 # Insert cars into invoicePosition
68 INSERT INTO invoiceposition (
69     InvoiceID, ProductType, ProductReferenceID,
70     Quantity, UnitNetPrice, VATRate
71 )
72 SELECT
73     new_InvoiceID,
74     'Car' AS ProductType,
75     rc.CarID AS ProductReferenceID,
76     rc.RentalDuration AS Quantity,
77     rc.DailyRateApplied AS UnitNetPrice,
78     input_VATRate
79 FROM rentalcar rc
80 WHERE rc.RentalID = input_RentalID;
81
82 # Insert amenities into invoicePosition
83 INSERT INTO invoiceposition (
84     InvoiceID, ProductType, ProductReferenceID,
85     Quantity, UnitNetPrice, VATRate
86 )
87 SELECT
88     new_InvoiceID,
89     'Amenity' AS ProductType,
90     ra.AmenityID AS ProductReferenceID,
91     ra.Quantity AS Quantity,
92     ra.DailyRateApplied AS UnitNetPrice,
93     input_VATRate
94 FROM rentalamenity ra
95 WHERE ra.RentalID = input_RentalID;
```

Najciekawsze funkcje - generuj fakturę część 4

```
97      # Calc totals for the invoice
98      SELECT
99          SUM(NetAmount) AS TotalNet,
100         SUM(VATAmount) AS TotalVAT,
101         SUM(GrossAmount) AS TotalGross
102      INTO
103         invoice_total_net, invoice_total_vat, invoice_total_gross
104      FROM invoiceposition
105      WHERE InvoiceID = new_InvoiceID;
106
107      # Update invoice with totals
108      UPDATE invoice
109      SET
110         TotalNetAmount = invoice_total_net,
111         TotalVAT = invoice_total_vat,
112         TotalGrossAmount = invoice_total_gross
113      WHERE InvoiceID = new_InvoiceID;
114
115      COMMIT;
116      END
```

Najciekawsze procedury

Dodaj rezerwację

```
1 CREATE DEFINER='herczyk_wiktoria_katarzyna'@`%` PROCEDURE `AddRental` (  
2     IN vRentalDate DATE,  
3     IN vExpectedReturnDate DATE,  
4     IN vVat DECIMAL(7,2),  
5     IN vReservationID INT,  
6     IN vDailyRate DECIMAL(7,2)  
7 )  
8 BEGIN  
9     DECLARE vTotalGrossAmount DECIMAL(8,2);  
10    DECLARE vDiscountApplied DECIMAL(7,2);  
11    DECLARE vNetAmount DECIMAL(8,2);  
12    DECLARE vRentalDuration SMALLINT(4);  
13    DECLARE vRentalID INT(10);  
14    DECLARE vCarID INT(10);  
15  
16    IF vReservationID not in (SELECT ReservationID from reservationcar) THEN  
17        SIGNAL SQLSTATE '45000'  
18        SET MESSAGE_TEXT = "Reservation doesn't exist";  
19    ELSEIF (SELECT ReservationStatus from reservation WHERE ReservationID = vReservationID) != "Active" THEN  
20        SIGNAL SQLSTATE '45000'  
21        SET MESSAGE_TEXT = "Reservation canceled or already completed";  
22    END IF;  
23  
24    start transaction;  
25  
26    SET vCarID = (SELECT COALESCE(CarID, NULL) from reservationcar where ReservationID = vReservationID);  
27    SET vRentalID = (SELECT COALESCE(MAX(RentalID), 0) + 1 FROM rental);  
28  
29    IF vDailyRate IS NULL THEN  
30        SET vDailyRate = (SELECT DailyRate FROM car WHERE CarID = vCarID);  
31    END IF;  
32  
33    SET vRentalDuration = (DATEDIFF(vExpectedReturnDate, vRentalDate) + 1);  
34    SET vNetAmount = (vDailyRate * vRentalDuration);  
35    SET vDiscountApplied = (SELECT CalculateDiscountAmount(vRentalDate, vExpectedReturnDate, vNetAmount));  
36  
37    IF vVat is NULL THEN  
38        Set vVat = 0.23 * vNetAmount;  
39    end if;  
40  
41  
42    SET vTotalGrossAmount = (vVat + vNetAmount - vDiscountApplied);  
43  
44    INSERT INTO rental (  
45        RentalID, RentalDate, ExpectedReturnDate, ActualReturnDate,  
46        TotalNetAmount, TotalVAT, TotalGrossAmount, InvoiceID,  
47        ReservationID, DiscountApplied  
48    ) VALUES (  
49        vRentalID, vRentalDate, vExpectedReturnDate, NULL,  
50        vNetAmount, vVat, vTotalGrossAmount, NULL,  
51        vReservationID, vDiscountApplied  
52    );  
53  
54    INSERT INTO rentalcar (  
55        RentalID, CarID, DailyRateApplied, DiscountApplied, RentalDuration  
56    ) VALUES (  
57        vRentalID, vCarID, vDailyRate, vDiscountApplied, vRentalDuration  
58    );  
59  
60    UPDATE reservation SET ReservationStatus = "Completed" WHERE ReservationID = vReservationID;  
61  
62    commit;  
63  
64    END
```

Inne ciekawe funkcjonalności

- Tabele audytowe
- Triggery
- Partycjonowanie
- Transakcje

Tabele audytowe - car_audit

AuditID # int(11)	CarID # int(11)	LicensePlateNumber # varchar(15)	VIN # varchar(17)	MakeID # int(11)	Model # varchar(50)	Year # smallint(6)	CarStatus # enum	DailyRate # decimal(7,2)	ChangeType # enum	ChangedBy # varchar(50)	ChangeTimestamp # timestamp
13	6	POZ11111	1HGCM82633A1	0	Golf	2015	Available	100,00	UPDATE	dylag_matylda_a	2025-01-26 19:53:40
4	6	POZ11111	1HGCM82633A1	0	Golf	2015	Available	100,00	UPDATE	herczyk_wiktoria	2025-01-22 09:33:44
27	20	WAW60606	1HGCM82633A6	0	Q5	2017	Under Maint	270,00	UPDATE	dylag_matylda_a	2025-01-26 19:53:58
25	18	KAT40404	1HGCM82633A4	0	3 Series	2018	Rented	250,00	UPDATE	dylag_matylda_a	2025-01-26 19:53:55
24	17	SZC30303	1HGCM82633A3	0	Astra	2020	Available	125,00	UPDATE	dylag_matylda_a	2025-01-26 19:53:54
23	16	GDA20202	1HGCM82633A2	0	Focus	2019	Available	130,00	UPDATE	dylag_matylda_a	2025-01-26 19:53:53
21	14	POZ99999	1HGCM82633A9	0	Clio	2016	Available	110,00	UPDATE	dylag_matylda_a	2025-01-26 19:53:50
20	13	GD88888	1HGCM82633A8	0	CX-5	2018	Under Maint	190,00	UPDATE	dylag_matylda_a	2025-01-26 19:53:49
19	12	WAW77777	1HGCM82633A7	0	Sportage	2020	Available	200,00	UPDATE	dylag_matylda_a	2025-01-26 19:53:48
17	10	KAT55555	1HGCM82633A5	0	208	2019	Available	140,00	UPDATE	dylag_matylda_a	2025-01-26 19:53:46
16	8	GDA33333	1HGCM82633A3	0	Civic	2018	Available	130,00	UPDATE	dylag_matylda_a	2025-01-26 19:53:44
14	7	LOD22222	1HGCM82633A2	0	Octavia	2016	Under Maint	110,00	UPDATE	dylag_matylda_a	2025-01-26 19:53:41
12	4	GD12345	1HGCM82633A3	0	A4	2017	Available	180,00	UPDATE	dylag_matylda_a	2025-01-26 19:53:40
11	3	WAW54321	1HGCM82633A5	0	X5	2020	Under Maint	300,00	UPDATE	dylag_matylda_a	2025-01-26 19:53:40
10	2	KRK98765	1HGCM82633A9	0	Fiesta	2019	Available	120,00	UPDATE	dylag_matylda_a	2025-01-26 19:53:39
9	1	KRA12345	1HGCM82633A1	0	Corolla	2018	Available	150,00	UPDATE	dylag_matylda_a	2025-01-26 19:53:38
7	20	WAW60606	1HGCM82633A6	0	Q5	2017	Under Maint	270,00	UPDATE	herczyk_wiktoria	2025-01-22 12:28:33
3	10	KAT55555	1HGCM82633A5	0	208	2019	Available	140,00	UPDATE	herczyk_wiktoria	2025-01-22 09:33:33
2	20	WAW60606	1HGCM82633A6	0	Q5	2017	Under Maint	270,00	UPDATE	herczyk_wiktoria	2025-01-22 09:33:15
1	8	GDA33333	1HGCM82633A3	0	Civic	2018	Available	130,00	UPDATE	dylag_matylda_a	2025-01-20 15:58:17
26	19	KRK50505	1HGCM82633A5	0	E-Class	2022	Under Maint	280,00	UPDATE	dylag_matylda_a	2025-01-26 19:53:57
22	15	LOD10101	1HGCM82633A1	0	Yaris	2021	Rented	140,00	UPDATE	dylag_matylda_a	2025-01-26 19:53:52
18	11	KRK66666	1HGCM82633A6	0	Qashqai	2021	Rented	220,00	UPDATE	dylag_matylda_a	2025-01-26 19:53:47

Triggery

AFTER UPDATE w tabeli car

```
BEGIN
  INSERT INTO `car_audit` ( `CarID`, `LicensePlateNumber`, `VIN`, `MakeID`, `Model`,
    `Year`, `CarStatus`, `DailyRate`, `ChangeType`, `ChangedBy`, `ChangeTimestamp` )
    VALUES
      (
        OLD.CarID,
        OLD.LicensePlateNumber,
        OLD.VIN,
        OLD.MakeID,
        OLD.Model,
        OLD.YEAR,
        OLD.CarStatus,
        OLD.DailyRate,
        'UPDATE',
        USER (),
        NOW()
      );
END
```

Triggery

AFTER DELETE w tabeli car

```
BEGIN
  INSERT INTO `car_audit` ( `CarID`, `LicensePlateNumber`, `VIN`, `MakeID`, `Model`
, `Year`, `CarStatus`, `DailyRate`, `ChangeType`, `ChangedBy`, `ChangeTimestamp` )
  VALUES
  (
    OLD.CarID,
    OLD.LicensePlateNumber,
    OLD.VIN,
    OLD.MakeID,
    OLD.Model,
    OLD.YEAR,
    OLD.CarStatus,
    OLD.DailyRate,
    'DELETE',
    USER (),
    NOW()
  );
END
```

Partycjonowanie

Partycjonowanie tabeli z autami na podstawie roku produkcji

Partition By:	<div>RANGE</div>	<div>`Year`</div>
Partitions:	<div>4</div>	
Subpartition By:	<div></div>	<div></div>
Subpartitions:	<div>0</div>	
Manually Create:	<input checked="" type="checkbox"/> Partition	<input checked="" type="checkbox"/> Subpartition

Partition Definition

<input checked="" type="checkbox"/> p_do_2015	2016
<input type="checkbox"/> p_do_2020	2021
<input type="checkbox"/> p_do_2025	2026
<input type="checkbox"/> p_max	MAXVALUE

Transakcje

Transakcja dodająca nową rezerwację

```
set transaction isolation level REPEATABLE read;  
start transaction;
```

IF

```
( CHAR_LENGTH( vPickupDate ) > 0 AND CHAR_LENGTH( vReturnDate ) > 0 AND CHAR_LENGTH( vCustomerID ) > 0 AND CHAR_LENGTH( vCarID ) > 0 ) THEN  
  INSERT INTO reservation ( ReservationID, PickupDate, ReturnDate, ReservationStatus, CustomerID )  
  VALUES  
    ( DEFAULT, vPickupDate, vReturnDate, "Active", vCustomerID );
```

```
SET x = ( SELECT MAX( ReservationID ) FROM reservation );  
INSERT INTO reservationcar ( ReservationID, CarID )  
VALUES  
  ( x, vCarID );
```

```
END IF;  
commit;
```

Dziękujemy za uwagę!