1. **Zdefiniuj pojęcia :**

**- dana**-informacja, fakt,

**- baza danych**-uporządkowany zbiór informacji, danych które są ze sobą wzajemnie powiązane i o wspólnych relacjach,

**- zakres bazy danych**-jest to określony fragment rzeczywistości który jest odwzorowany,

**- system zarządzania bazami** **danych**-zespół narzędzi do zarządzania bazami danych w celu: utrzymanie, pielęgnacja, tworzenie, modyfikacja, zarządzania uprawnieniami i użytkownikami

**- system baz danych**-zespół baz danych z wszystkimi elementami: SBD [Programy użytkowe, aplikacje, DBMS (Przetwarzanie zapytań, Operacje na zapamiętanych bazach danych), Dane systemowe, rzeczywista baza danych]

1. **Opisz proces modelowania danych. Scharakteryzuj poszczególne fazy.**

- **Określenie zakresu odwzorowanej rzeczywistości,** (jaki ma zasięg, jak najlepiej odzwierciedlić rzeczywistość) Wytypowanie obiektów (rzeczywistych i abstrakcyjnych) do rzeczywistego odwzorowania w bazie danych.

- **Analiza i lista wymagań stawianych modelowi** ( jakie wymagania będzie miał użytkownik , jakie parametry) jakie będą występować bezpośrednie powiązania między obiektami.

- **Projekt koncepcyjny** (budowanie semantycznego modelu danych; techniki ERD lub SERM, diagramu związków NC, stworzenie czystego modelu bez obciążenia danymi)określenie sposobu realizacji mając na uwadze złożoność i stopień relacji. Pokazanie jakie cechy charakteryzują te powiązania, tworzenie listy atrybutów, konkretnych obiektów NC

- **Projekt logiczny** (rozszerzenie modelu semantycznego o implementację na serwerze, typy danych)

**- Realizacja fizyczna** (fizyczne instalowanie bazy na serwerze za pomocą CASE, uruchomienie skryptu

1. **Zdefiniuj encję i relację w diagramie związków encji (ERD).**

- **encja**; jest to reprezentacja wyobrażonego lub rzeczywistego obiektu który posiada następujące cechy: jest unikalny, jednoznacznie określony ( jest rzeczownikiem oraz jest w diagramie prostokątem)

- relacja; określa związki między encjami, najczęściej występuje między dwoma encjami.

1. **Co to jest atrybut? Przedstaw klasyfikacje atrybutów.**

- atrybut jest pewną własnością encji, o której chcemy przechowywać informacje i jest reprezentowany przez tą własność.

- atrybuty możemy sklasyfikować jako; - proste (nazwisko) i złożone (pełny adres), - jednowartościowe (data) i wielowartościowe (rachunek, adres, telefon, e-mail), - zapamiętane (zapisane w bazie) i wyznaczone (wyliczone na podstawie zapamiętanych (wiek na podstawie daty urodzenia)

1. **Co to jest klucz główny i jakie posiada własności?**

- klucz główny jest to atrybut lub zbiór atrybutów jednoznacznie identyfikujący dany egzemplarz encji, jest unikalny i nie może mieć wartości NULL

1. **Co to jest stopień relacji?**

- stopień relacji określa ilość typów encji wchodzących w skład jednej relacji, (najczęściej są dwie)

1. **Wymień i opisz przypadki wystąpienia złożoności relacji w notacji (1,M,N).**

- (1:M:N) – zawiera rodzaje relacji (1:1), (1:N), (M:N). (1:1) jedna encja typu A może wejść w relację z tylko jedną encją typu B z drugiego zbioru (kobieta w związku małżeńskim z jednym mężczyzną). (1:N) Encja A może wejść w relację z wieloma encjami B. Encja B tylko z jedną ncją typu A (ojciec ;dzieci). (M:N) wiele do wielu (język:student).

1. **Podaj różnicę pomiędzy typami E w modelu ERD i SERM.**

W modelu ERD encja jest słabym typem encji, jej istnienie zależy od innych typów encji, posiadają układ dowolny, klucz dowolny, czytelny

W modelu SERM istnienie ncji nie zależy od żadnej innej ncji, posiadają ściśle określoną strukturę hierarchii, klucz zgodny z hierarchią dziedziczenia

1. **Podaj różnicę pomiędzy typami R w modelu ERD i SERM.**

W modelu ERD nie wszystkie relacje R mają odwzorowanie w bazie

W modelu SERM każdy obiekt R ma odwzorowanie w bazie

1. **Zdefiniuj typ ER w modelu SERM.**

- Encjo-relacja ER jest typem encji który ma swoje atrybuty kluczowe oraz obce atrybuty kluczowe, jest złożonym typem encji w której występuje relacja encji (1:1), może posiadać obiekt NULL,

1. **Wymień i opisz przypadki wystąpienia złożoności relacji w notacji (min;max).**

- notacja (min, max) jest to relacja (1,M,N)(1:1, 1:M, 1:N) czyli minimalna ilość wystąpień relacji (0,1) i maxymalna ilość wystąpień relacji (1,\*)

(0,1)(0,\*)(1,1)(1,\*)

1. **Opisz sposoby dziedziczenia dostępne w modelu SERM.**

- dziedziczenie w relacjach polegana tym że obiekt rodzic, po lewej stronie a dziedziczący po prawej stronie, obiekt dziedziczący dziedziczy klucz główny od obiektu rodzica (z lewej strony) obiekt z lewej strony przekazuje klucz główny do prawegoobiektu.

1. **Kiedy atrybut modelujemy jako nową encję?**

- gdy chcemy pamiętać dynamiczny zbiór wartości danego atrybutu, niezależnie od istnienia opisywanego obiektu.

- gdy atrybut sam jest opisany dodatkowymi atrybutami

- gdy atrybut dla jednego wystąpienia encji może przyjąć wiele wartości.

1. **Kiedy zamiast związku (M:N) wstawiamy nową encję?**

- gdy fakt modelowania związkiem (relacją) wymaga opisu za pomocą dodatkowych atrybutów

1. **Podaj własności zastępowania związku (M:N) nową encją.**

-

1. **Jak modelujemy atrybuty wielowartościowe?**

- **atrybuty wielowartościowe** są to atrybuty które posiadają wiele wartości na kilka nr telefonów, nr rachunków,

1. **Jak modelujemy atrybuty złożone?**

**- atrybuty złożone** są to atrybuty na które składa się kilka powiązanych informacji, np. atrybut : adres na który składa się : miasto, kod pocztowy, ulica, nr posesji.

1. **Jak modelujemy agregację?**

-

1. **Co to jest SQL?**

-to skrót oznaczający **Structured Query Language** (strukturalny język zapytań).

SQL to standardowy język dostępu i zarządzania bazami danych. Poprzez komendy SQL-a realizowane są m.in. takie operacje jak:

definicja i modyfikacja obiektów baz danych

dostęp i aktualizacja bazy danych

czynności administracyjne

tworzenie procedur składowanych

1. **Jakie operacje wykonywane są poprzez komendy SQL-a?**

definiowanie i modyfikacja obiektów bazy danych

wyszukiwanie i aktualizacja bazy danych

administrowanie bazą danych.

1. **SQL w klasyfikacji relacyjnych języków zapytań.**

**Język SQL** należy do grupy **języków nieproceduralnych** i **deklaratywnych**, zatem daje użytkownikowi możliwość określenia (deklaracji) potrzeb bez podawania sposobu realizacji. Problem metody rozwiązania został przerzucony na SZBD.

1. **Przedstaw znane Ci formy SQL-a.**

Dostęp do bazy danych realizowany jest przez język SQL w dwóch trybach:

**interaktywnym** (ang. Interactive SQL)– instrukcje wprowadzane są interakcyjnie (spoza środowiska aplikacji) bezpośrednio do SZBD, który natychmiast prezentuje wyniki odpowiedzi

**zanurzonym** (ang. Embedded SQL)- instrukcje umieszczone są wewnątrz konstrukcji języka programowania aplikacji na dwa

sposoby:

**- kod statyczny** (ang. Static SQL) – stały kod tworzony przy projektowaniu aplikacji przez jej projektanta

**- kod dynamiczny** (ang. Dynamic SQL) – jest odpowiedzią na działanie użytkownika a nie projektanta aplikacji, zatem generowany jest w trakcie działania aplikacji

1. **Zdefiniuj składniki SQL-a.**

Zbiory instrukcji poszczególnych grup funkcjonalnych stanowią odpowiednio:

język definiowania danych (ang. Data Definition Language DDL)

język manipulowania danymi (ang. Data Manipulation Language DML)

język kontroli danych (ang. Data Control Language DCL)

1. **Wymień znane Ci instrukcje DML-a.**

Język DML stanowi zbiór instrukcji, które operują na danych w bazy danych:

**Select** – wyszukiwanie danych

**Insert** – tworzenie nowego rekordu w tablicy

**Update** – tworzenie lub modyfikacja danych w istniejącym rekordzie

**Delete** – usunięcie rekordu w tablicy

1. **Wymień znane Ci instrukcje DCL-a.**

Język DCL stanowi zbiór instrukcji, które odpowiadają czynnościom administracyjnym:

Describe – opis struktury zadanej tablicy

Grant/Revoke – nadawanie/usuwanie uprawnień

Connect – ustanowienie połączenia

Commit/Rolback– zatwierdzenie/cofnięcie transakcji

Open/Close – otwieranie/zamykanie kursora

Execute – wykonanie przygotowanego kodu

1. **Z czego składają się polecenia SQL-a?**

Polecenia języka SQL składają się z dwóch typów słów:

**zarezerwowanych**, które stanowią integralną część języka i nie mogą być modyfikowane ani dzielone pomiędzy wierszami; słowa kluczowe zwykle pisane są wielkimi literami (SELECT, GROUP BY, CREATE VIEW itp.)

**zdefiniowanych przez użytkownika**, które reprezentują nazwy różnych, stworzonych przez użytkownika, obiektów pisanych zwyczajowo małymi literami

W języku SQL występują dwa typy instrukcji:

polecenia ujęte w standardzie

rozszerzenia standardu – dodatkowe polecenia typowe dla konkretnego systemu zarządzania bazami Kod SQL wykorzystujący rozszerzenia specyficzne dla wybranego systemu zarządzania bazami danych nie jest przenaszalny!

1. **Wymień obiekty SQL-a.**

**Database** - baza danych

**Table** – tablela (relacja) w bazie danych

**Column** – kolumna (atrybut) tabeli

**Index** – tablica indeksów do tabeli

**View** – perspektywa, widok tabeli lub tabel

**Synonym** – alternatywna nazwa tabeli lub widoku

**Schema** – schemat, grupa powiązanych ze sobą obiektów, posiadająca nazwę i znajdująca się pod kontrolą jednego identyfikatora uprawnień

**Domena** – określa typ danych, logiczne kategorie porównywalnych wartości

1. **Opisz definicję danych w SQL-u.**

Różnorodność typów danych i możliwość określania ich długości służy **optymalizacji definicji tabeli** pod kątem **zużycia miejsca** i do **kontroli integralności** wprowadzanych lub wynikowych wartości.

Do **utworzenia nowej tabeli** służy instrukcja **CREATE TABLE**, która wymaga podania nazwy tabeli, wszystkich kolumn (nazwa kolumny, typ danych, opcjonalnie maksymalna długość danych w kolumnie i opcja kolumny) oraz ewentualnych opcji.

1. **Scharakteryzuj tekstowy typ danych w SQL-u.**

Typy tekstowe:

**CHAR (N)** – pole o stałej długości N znaków (ew. uzupełniane spacjami)

**VARCHAR (N)** – pole o zmiennej długości nie przekraczającej N znaków

**BIT (N)** – pole o stałej długości N bitów

**BIT VARING (N)** - pole o zmiennej długości nie przekraczającej N bitów

1. **Scharakteryzuj liczbowy typ danych w SQL-u.**

Typy liczbowe:

- różniące się zakresem (zależnym od SZBD)

**INT** lub **INTEGER** – liczba całkowita

**BIGINT** – (duża) liczba całkowita o większej liczbie cyfr

**SMALLINT** – (mała) liczba całkowita o mniejszej liczbie cyfr

- różniące się precyzją

**REAL** lub **FLOAT** – liczba rzeczywista zmiennoprzecinkowa

**DOUBLE PRECISION** - liczba zmiennoprzecinkowa

**DECIMAL(M,N)** lub **NUMERIC(M,N)** – liczba stałoprzecinkowa o M cyfrach znaczących i N miejscach po przecinku

1. **Co to jest Interval?**
2. **Przedstaw opcje kolumn dostępne w instrukcjach DDL-a.**

**NULL/NOT NULL** – pole w tej kolumnie może/nie może przyjmować wartości pustej (NULL)

**UNIQUE** – wartość pola musi być unikatowa w kolumnie

**PRIMARY KEY** – kolumna stanowi klucz główny tabeli

**REFERENCES [nazwa\_tablicy]** – kolumna stanowi klucz obcy wskazujący na podaną tablicę

**DEFAULT** wartość\_domyślna – wszystkie pola kolumny wypełnione są wartością domyślną

**[NOT] CASE-SENSITIVE** – określa, czy przy porównywaniu wartości w kolumnie znakowej mają być rozróżniane małe i wielkie litery

**AUTOINCREMENT** – automatyczne nadawanie wartości pola w kolumnie podczas wstawianie nowego wiersza

**FORMAT format** –wartości pól w kolumnie wyświetlane będą w podanym formacie

**LABEL etykieta** – nadanie etykiety (nagłówka) kolumnie przy wyświetlaniu wartości jej pól

UWAGA! Opcja PRIMARY KEY automatycznie włącza opcje NOT NULL i UNIQUE!

1. **Jakie znasz opcje modyfikacji schematu?**

Opcje modyfikacji schematu dotyczą modyfikacji kolumn w zadanej tablicy:

**ADD COLUMN** nazwa\_kolumny typ\_danych [(długość) opcja\_kol] - dodanie nowej kolumny do tablicy wymaga podanie parametrów i opcji kolumny takich jak przy tworzeniu tablicy

**DROP COLUMN** nazwa\_kolumny - usunięcie kolumny o podanej nazwie

**ALTER COLUMN** nazwa\_kolumny [opcja\_kol] - modyfikacja kolumny o podanej nazwie

1. **Opisz opcje dostępne w parametrach projekcji instrukcji SELECT.**

**Opcje projekcji** to parametry operacji rzutowania („cięcia pionowego”) na tabelach wymienionych po frazie FROM (definicja kolumn tabeli wynikowej).

- Stanowią wybrane lub wszystkie (\*) nazwy kolumn i/lub działające na nich **funkcje agregujące** oraz **operatory.**

- Powtarzające się nazwy kolumn pochodzących z różnych tabel poprzedza się nazwą tabeli (nazwa\_tabeli.nazwa\_kolumny) lub jej aliasem (alias.nazwa\_kolumny)

- Występują z następującymi opcjami:

- klauzula **DISTINCT** ignoruje duplikaty wartości we wskazanej kolumnie w przeciwieństwie do domyślnej klauzuli **ALL** akazującej wybrać wszystkie wiersze

- klauzula **TOP n [PERCENT]** tworzy listę tylko n początkowych wierszy lub n procent początkowych wierszy

- klauzula **AS** po nazwie kolumny tworzy jej alias

1. **Jakie znasz funkcje agregujące i gdzie w instrukcji SELECT można ich używać?**

W języku SQL dostępne są następujące funkcje agregujące:

**count** () – zwraca liczność zbioru

**count** ([DISTINCT] nazwa kolumny) – zwraca liczbę wierszy we wskazanej kolumnie

**sum** ([DISTINCT] nazwa kolumny) – zwraca sumę wartości podanej kolumny

**avg** ([DISTINCT] nazwa kolumny) – zwraca średnią wartość podanej kolumny

**min** ([DISTINCT] nazwa kolumny) – zwraca minimalną wartość podanej kolumny

**max** ([DISTINCT] nazwa kolumny) – zwraca maksymalną wartość podanej kolumny

Funkcje **sum()** i **avg()** dostępne są tylko dla typów liczbowych, a dla **min()** i **max()** kolumna może być także typu znakowego lub daty.

1. **Przedstaw operatory specjalne dostępne we frazie WHERE.**

operator specjalne:

**- IN** - równy wartości z listy podanej przez użytkownika (odpowiada przyrównaniom do różnych wartości połączonych operatorem OR) lub będącej wynikiem instrukcji SELECT

**- BETWEEN** – równy jednej z wartości z podanego przedziału (odpowiada dwóm wyrażeniom: z operatorem „>” i dolną granicą przedziału oraz z operatorem „<” i górną granicą przedziału połączonym operatorem AND)

**- LIKE** - równy z podanym wzorcem (dla kolumn tekstowych), który może zawierać **symbole wieloznaczne**: **\_** (podkreślenie) oznacza

dopasowanie do dowolnego pojedynczego znaku, **%** oznacza dopasowanie do dowolnej liczby dowolnych znaków

**- IS NULL** - równy wartości pustej

1. **Do czego służy klauzula HAVING?**

HAVING – warunek nałożony na grupę.

1. **Jaki jest domyślny porządek prezentacji wyników instrukcji SELECT?**

**Parametry porządkowania wyników** – stanowią wybrane lub wszystkie nazwy kolumn lub numer(y) kolumn(y) oznaczający pozycję kolumny na liście opcji projekcji oraz opcje porządkujące kolejność wyświetlania wierszy wynikowych.

- domyślnie przyjęty jest porządek rosnący **ASC** w przeciwieństwie do porządku malejącego oznaczanego opcją **DESC**

**38. Składnia SQL**

Język SQL nie rozróżnia małych i dużych liter w słowach kluczowych i nazwach (baz danych, tabel, indeksów i kolumn)

- Nazwy w SQL-u składają się ze znaków alfanumerycznych, nie rozpoczynających się od cyfr. Nazwa nie może składać się z samych cyfr lub zawierać znaki przestankowe bądź „@”

- Każda komenda SQL-a kończy się średnikiem (;) i może składać się z wielu linii tekstu

- Wartości tekstowe podaje się w cudzysłowie (”napis”) lub w apostrofach (’napis’)

- Wartości liczbowe podaje się bezpośrednio; z kropką dziesiętną (liczby dziesiętne) lub w notacji wykładniczej (liczby iennoprzecinkowe)

- SQL **nie zawiera** zwykłych **instrukcji sterowania** typu: IF…THEN, DO…WHILE itp.