Otázky do diskusie

Popíšte jednotlivé typy pripojení, zakreslite ich schematicky a uveďte príklady. Popíšte rozdelenie väzby n:m na jednotlivé časti a uveďte príklady. Popíšte základné prvky ERM. Popíšte základné rozdelenie jazyka SQL a uveďte niektoré príkazy k každej časti. Popíšte čo je to primárny a cudzí kľúč, a index. Popíšte vzťah medzi pojmami transakcia, konzistentnosť, redundancia. Popíšte úlohu Azure Elastic Database Pool, Popíšte možnosti fragmentácie databázy a uveďte konkrétny príklad. Popíšte odlišnosti poskytovania PaaS poskytovanie databázy pre Azure SQL a Amazon AWS RDS.

Popíšte jednotlivé typy pripojení, zakreslite ich schematicky a uveďte príklady.

Existuje mnoho rôznych typov pripojení, ktoré sa používajú na prepojenie rôznych zariadení a sieťových prvkov. Niektoré z týchto typov pripojenia zahŕňajú:

1. USB (Universal Serial Bus) - Toto pripojenie umožňuje prenos dát medzi zariadeniami. USB porty sa často nachádzajú na počítačoch, tlačiarňach, telefónoch, tabletových počítačoch a iných zariadeniach. Existuje niekoľko typov USB portov, ktoré sa líšia podľa rýchlosti prenosu dát a veľkosti konektoru.
2. Ethernet - Ethernet je typ pripojenia, ktorý sa používa na prenos dát v lokálnej sieti (LAN). Ethernet pripojenie umožňuje pripojenie zariadení ako počítačov, tlačiarní, serverov a iných sieťových prvkov. Ethernet pripojenie používa štandardné káble s RJ-45 konektormi a existuje niekoľko rýchlostí prenosu dát Ethernetu, vrátane 10 Mbps, 100 Mbps a 1 Gbps.
3. Wi-Fi - Wi-Fi je bezdrôtový typ pripojenia, ktorý umožňuje prenos dát medzi zariadeniami pomocou rádiových vĺn. Wi-Fi pripojenie sa používa na pripojenie mobilných zariadení, ako sú smartfóny, tablety a notebooky, k internetu a lokálnym sieťam.
4. Bluetooth - Bluetooth je bezdrôtový typ pripojenia, ktorý sa používa na prenos dát medzi zariadeniami na krátke vzdialenosti. Bluetooth pripojenie sa používa na prepojenie slúchadiel, reproduktorov, klávesníc a myší s mobilnými zariadeniami a počítačmi.
5. HDMI (High-Definition Multimedia Interface) - HDMI pripojenie sa používa na prenos obrazu a zvuku z jedného zariadenia do druhého. HDMI káble sa používajú na pripojenie televízorov, monitory, herných konzol a iných zariadení k počítačom a iným zdrojom obrazu.
6. DisplayPort - DisplayPort je ďalší typ pripojenia, ktorý sa používa na prenos obrazu a zvuku. DisplayPort káble sa používajú na pripojenie monitorov a iných zariadení k počítačom a iným zdrojom obrazu. DisplayPort má viac funkcií ako HDMI a umožňuje vyššie rozlíšenie a rýchlejší prenos dát.
7. Thunderbolt - Thunderbolt je rýchle a všestranný typ prip

Existuje niekoľko typov pripojení v databázach, ktoré sa používajú na prepojenie dátových tabuliek alebo záznamov v rôznych tabuľkách. Niektoré z týchto typov pripojenia zahŕňajú:

1. One-to-One - Jednoznačné pripojenie je typ pripojenia, ktorý prepojí jeden záznam v jednej tabuľke s jedným záznamom v inej tabuľke. Tento typ pripojenia sa často používa na prepojenie tabuliek súvisiacich s používateľmi alebo objednávkami, kde každý záznam v jednej tabuľke sa odkazuje na presne jeden záznam v inej tabuľke. Príkladom môže byť prepojenie tabuliek používateľov a ich fakturačných údajov, kde každý záznam v tabuľke používateľov sa odkazuje na presne jeden záznam v tabuľke fakturačných údajov.
2. One-to-Many - Jednoznačné pripojenie je typ pripojenia, ktorý prepojí jeden záznam v jednej tabuľke s viacerými záznamami v inej tabuľke. Tento typ pripojenia sa často používa na prepojenie tabuliek súvisiacich s objednávkami a položkami objednávky, kde každý záznam v tabuľke objednávok sa odkazuje na viacero záznamov v tabuľke položiek objednávky. Príkladom môže byť prepojenie tabuliek objednávok a položiek objednávky, kde každý záznam v tabuľke objednávok sa odkazuje na viacero záznamov v tabuľke položiek objednávky.
3. Many-to-Many - Mnoho-ku-mnohým pripojenie je typ pripojenia, ktorý prepojí viacero záznamov v jednej tabuľke s viacerými záznamami v inej tabuľke. Tento typ pripojenia sa používa na prepojenie tabuliek, ktoré môžu obsahovať viacero záznamov, ktoré sa odkazujú na viacero záznamov v inej tabuľke. Príkladom môže byť prepojenie tabuliek používateľov a záujmov, kde každý používateľ môže mať viacero záujmov a každý záujem môže byť priradený k viacerým používateľom.

opíšte rozdelenie väzby n:m na jednotlivé časti a uveďte príklady.

Väzba mnoho-ku-mnohým (n:m) je typ vzťahu medzi dvoma entitami, v ktorom každá entita môže byť spojená s viacerými inými entitami. Rozdelenie väzby n:m sa dá rozdeliť na tri časti:

1. Entita A - reprezentuje prvú entitu v mnoho-ku-mnohým väzbe.
2. Entita B - reprezentuje druhú entitu v mnoho-ku-mnohým väzbe.
3. Tabuľka prepojení - táto tabuľka uchováva vzťahy medzi entitami A a B. Táto tabuľka obsahuje minimálne dva stĺpce, jeden stĺpec pre identifikátor A a druhý stĺpec pre identifikátor B.

Príkladom môže byť vzťah medzi tabuľkami používateľov a záujmov, kde každý používateľ môže mať viacero záujmov a každý záujem môže byť priradený k viacerým používateľom. V tomto prípade by tabuľka prepojení mohla obsahovať dva stĺpce: jednu stĺpec pre identifikátor používateľa a druhý stĺpec pre identifikátor záujmu.

Ďalším príkladom môže byť vzťah medzi tabuľkami kníh a autorov, kde každá kniha môže mať viacerých autorov a každý autor môže mať napísaných viacero kníh. V tomto prípade by tabuľka prepojení mohla obsahovať dva stĺpce: jeden stĺpec pre identifikátor knihy a druhý stĺpec pre identifikátor autora.

Rozdelenie väzby n:m na tieto tri časti umožňuje efektívne ukladanie a získavanie dát z databázy, pretože umožňuje zachovanie správnej štruktúry pre všetky entitné typy a ich vzájomné prepojenia.

Popíšte základné prvky ERM

ERM (Entity-Relationship Model) je konceptuálny model používaný na návrh a popis relačných databáz. Základné prvky ERM zahŕňajú:

1. Entitu: Entita predstavuje objekt alebo vec, o ktorej sa ukladajú údaje do databázy. Môže to byť napríklad zákazník, produkt, zamestnanec atď. Každá entita má svoje vlastné atribúty, ktoré opisujú charakteristiky danej entity.
2. Vzťahy medzi entitami: Vzťahy predstavujú spôsoby, akými sa entity navzájom prepojili. Môžu to byť napríklad vzťahy "jeden-na-jeden", "jeden-na-viac" alebo "viac-na-viac". Tieto vzťahy sa v ERM reprezentujú pomocou šípok, ktoré ukazujú, ako sú entitami spojené.
3. Atribúty: Atribúty sú charakteristiky entít, ktoré opisujú ich vlastnosti. Môžu to byť napríklad meno, adresa, vek, číslo účtu a podobne.
4. Kľúče: Kľúče sú atribúty, ktoré slúžia na identifikáciu jedinečnej entity v databáze. Kľúče môžu byť primárne alebo cudzie, pričom primárny kľúč je kľúč, ktorý jednoznačne identifikuje každú entitu v databáze.
5. Domény: Domény predstavujú množinu možných hodnôt, ktoré môžu byť priradené k atribútu. Doména môže byť napríklad reťazec znakov, číslo, dátum, čas a podobne.
6. Diagramy ERM: Diagramy ERM sú vizuálne zobrazenia entít, vzťahov a atribútov, ktoré sú použité v databáze. Diagramy ERM umožňujú ľahšie porozumenie štruktúry databázy a môžu slúžiť ako základ pre fyzický návrh databázy.

Popíšte základné rozdelenie jazyka SQL(ddl,dml atd.) a uveďte niektoré príkazy k každej časti.

SQL (Structured Query Language) je jazyk používaný na prácu s relačnými databázami. SQL sa často delí na dve hlavné časti:

1. DDL (Data Definition Language) - slúži na definovanie databázových objektov, ako sú tabuľky, stĺpce a indexy.

Niektoré základné príkazy DDL zahŕňajú:

* CREATE TABLE: vytvára novú tabuľku v databáze
* ALTER TABLE: umožňuje zmeniť štruktúru existujúcej tabuľky, napríklad pridať alebo odstrániť stĺpce
* DROP TABLE: odstraňuje existujúcu tabuľku z databázy
* CREATE INDEX: vytvára index na stĺpec alebo skupinu stĺpcov v tabuľke, aby sa zlepšila výkonosť vyhľadávania

1. DML (Data Manipulation Language) - slúži na manipuláciu s dátami v tabuľkách. S DML môžeme vkladať, aktualizovať a mazať dáta.

Niektoré základné príkazy DML zahŕňajú:

* SELECT: vyberá dáta z jednej alebo viacerých tabuliek
* INSERT: vkladá nové dáta do tabuľky
* UPDATE: aktualizuje existujúce dáta v tabuľke
* DELETE: mazá dáta z tabuľky

Okrem DDL a DML existujú aj ďalšie časti SQL, ako napríklad DCL (Data Control Language) a TCL (Transaction Control Language), ktoré slúžia na správu prístupu k dátam a na riadenie transakcií.

SQL (Structured Query Language) je jazyk používaný na prácu s relačnými databázami. SQL sa často delí na niekoľko častí:

1. Data Definition Language (DDL) - slúži na definovanie databázových objektov, ako sú tabuľky, stĺpce a indexy. Niektoré základné príkazy DDL zahŕňajú:

* CREATE TABLE: vytvára novú tabuľku v databáze
* ALTER TABLE: umožňuje zmeniť štruktúru existujúcej tabuľky, napríklad pridať alebo odstrániť stĺpce
* DROP TABLE: odstraňuje existujúcu tabuľku z databázy
* CREATE INDEX: vytvára index na stĺpec alebo skupinu stĺpcov v tabuľke, aby sa zlepšila výkonosť vyhľadávania

1. Data Manipulation Language (DML) - slúži na manipuláciu s dátami v tabuľkách. S DML môžeme vkladať, aktualizovať a mazať dáta. Niektoré základné príkazy DML zahŕňajú:

* SELECT: vyberá dáta z jednej alebo viacerých tabuliek
* INSERT: vkladá nové dáta do tabuľky
* UPDATE: aktualizuje existujúce dáta v tabuľke
* DELETE: mazá dáta z tabuľky

1. Data Control Language (DCL) - slúži na správu prístupu k dátam a kontrolu bezpečnosti. Niektoré základné príkazy DCL zahŕňajú:

* GRANT: umožňuje prideliť určité práva pre prístup k databáze alebo tabuľke
* REVOKE: zruší určité práva pre prístup k databáze alebo tabuľke

1. Transaction Control Language (TCL) - slúži na riadenie transakcií. Niektoré základné príkazy TCL zahŕňajú:

* COMMIT: ukončí aktuálnu transakciu a potvrdí všetky zmienky v databáze
* ROLLBACK: zruší všetky zmeny, ktoré sa vykonali v rámci aktuálnej transakcie a vráti databázu do jej predchádzajúceho stavu

Každá časť jazyka SQL slúži na iný účel a umožňuje vykonávať rôzne operácie v relačnej databáze.

Popíšte čo je to primárny a cudzí kľúč, a index.

Primárny kľúč je unikátny identifikátor pre každý záznam v databáze. Tento kľúč sa používa na rýchle vyhľadávanie a usporiadanie záznamov v databáze. Každá tabuľka v databáze má práve jeden primárny kľúč.

Cudzí kľúč je kľúč, ktorý slúži na prepojenie dvoch tabuliek v databáze. Cudzí kľúč je zvyčajne kľúčom v jednej tabuľke a odkazuje na primárny kľúč v inej tabuľke. Týmto spôsobom sa vytvárajú vzťahy medzi tabuľkami a zabezpečuje sa integrita dát.

Index je štruktúra dát, ktorá umožňuje rýchle vyhľadávanie záznamov v databáze. Indexy sa vytvárajú na základe jedného alebo viacerých stĺpcov v tabuľke a zlepšujú výkon vyhľadávania dát v databáze. Indexy sa používajú na vyhľadávanie, zoradenie a filtrovanie záznamov v databáze. Všetky dotazy, ktoré obsahujú podmienku, ktorá sa zhoduje so stĺpcom v indexe, budú výrazne rýchlejšie ako dotazy bez indexu.

Popíšte vzťah medzi pojmami transakcia, konzistentnosť, redundancia.

Transakcia, konzistentnosť a redundancia sú v oblasti databázových systémov kľúčovými pojmy, ktoré sú úzko prepojené a vzájomne ovplyvňujú jeden druhého.

Transakcia je základnou jednotkou spracovania dát v databázových systémoch. Predstavuje sériu operácií, ktoré majú byť vykonané ako celok - buď všetky operácie uspejú a transakcia je potvrdená (commit), alebo aspoň jedna operácia zlyhá a transakcia je zrušená (rollback). Cieľom transakcií je zabezpečiť konzistentnosť databázy, t.j. aby dáta v databáze zostali v konzistentnom stave, aj keď sa vyskytnú chyby pri vykonávaní transakcií.

Konzistentnosť sa vzťahuje k tomu, že dáta v databáze musia byť v každom momente v konzistentnom stave, t.j. že musia spĺňať všetky pravidlá a obmedzenia, ktoré sú na ne aplikované. Konzistentnosť je zabezpečovaná pomocou transakcií, ktoré zabezpečujú, že všetky operácie vykonané v rámci jednej transakcie budú buď úspešné alebo zrušené. Týmto spôsobom sa zabezpečuje, že dáta zostanú v konzistentnom stave a že sa nevyskytnú nekonzistentné stavy.

Redundancia znamená, že v databáze sú uložené dáta viackrát, napríklad v rôznych tabuľkách alebo v rôznych stĺpcoch. Cieľom redundancie je zabezpečiť, aby v prípade chyby alebo havárie boli dáta chránené a bolo možné obnoviť databázu zo záloh. Redundancia môže však viesť aj k problémom s konzistentnosťou, ak sa dáta v rôznych miestach líšia a nie je jasné, ktoré sú správne. Preto je dôležité, aby bolo v databáze zabezpečené konzistentné ukladanie dát a aby redundancia bola použitá s opatrnosťou.

Vzťah medzi týmito troma pojmami spočíva v tom, že transakcie zabezpečujú konzistentnosť databázy a redundancia slúži na ochranu dát pred strátou. Pre dosiahnutie optimálneho stavu je dôležité nájsť správnu rovnováhu medzi konzistentnosťou a redundanciou, aby sa minimalizovali problémy

Popíšte úlohu Azure Elastic Database Pool, Popíšte možnosti fragmentácie databázy a uveďte konkrétny príklad.

Azure Elastic Database Pool je služba v cloude od spoločnosti Microsoft, ktorá umožňuje správu a škálovanie sady databáz v Azure SQL Database. Táto služba umožňuje spravovať a optimalizovať viacero databáz v rámci jednej sady, čo umožňuje lepšiu využiteľnosť zdrojov a znižuje náklady na správu a prevádzku.

Azure Elastic Database Pool poskytuje dynamické pridelenie prostriedkov pre každú databázu v rámci sady, v závislosti na aktuálnych potrebách. To znamená, že keď databáza potrebuje viac zdrojov, ako je momentálne pridelené, systém automaticky pridá ďalšie zdroje na zabezpečenie optimálneho výkonu.

Fragmentácia databázy sa používa na rozdelenie veľkej databázy na menšie časti, ktoré sa nazývajú fragmenty alebo oddiely. Toto umožňuje lepšie využitie zdrojov a rýchlejší prístup k dátam. Existujú tri základné typy fragmentácie databázy:

1. Horizontálna fragmentácia - tento typ fragmentácie rozdeľuje dáta podľa riadkov. Každý riadok v databáze patrí do jedného z fragmentov. Táto metóda sa často používa na rozdeľovanie dát podľa geografickej polohy alebo iných kritérií.
2. Vertikálna fragmentácia - táto metóda fragmentuje databázu podľa stĺpcov. Každý fragment obsahuje určitý počet stĺpcov. Táto metóda sa často používa na rozdeľovanie dát podľa citlivosti alebo prístupu.
3. Hybridná fragmentácia - táto metóda kombinuje horizontálnu a vertikálnu fragmentáciu.

Konkrétny príklad fragmentácie databázy môže byť napríklad e-commerce platforma, ktorá ukladá zákaznícku objednávku, zákaznícku historiu a produkty v rôznych databázach. Horizontálna fragmentácia môže byť použitá na rozdeľovanie objednávok podľa zákazníka alebo podľa geografického umiestnenia. Vertikálna fragmentácia môže byť použitá na rozdeľovanie citlivých informácií, ako sú informácie o platbe, do samostatnej databázy. Takéto rozdelenie umožňuje lepšie využitie zdrojov a zlepšuje rýchlosť prístupu k

Popíšte odlišnosti poskytovania PaaS poskytovanie databázy pre Azure SQL a Amazon AWS RDS.

PaaS (Platform-as-a-Service) poskytuje užívateľom plne spravované prostredie pre vývoj, testovanie a nasadenie aplikácií bez nutnosti spravovať infraštruktúru a hardvér. Azure SQL a Amazon RDS sú dva príklady PaaS služieb, ktoré ponúkajú správu databáz.

Azure SQL a Amazon RDS sú si v mnohých ohľadoch podobné, ale existujú aj niektoré zásadné odlišnosti:

1. Typ databázy: Azure SQL podporuje iba SQL Server databázy, zatiaľ čo Amazon RDS podporuje rôzne typy databáz, vrátane MySQL, PostgreSQL, Oracle, SQL Server a ďalších.
2. Správa: Azure SQL je plne spravovaná služba, ktorá neumožňuje používateľom prístup k operačnému systému alebo inštalovať vlastný softvér. Na druhej strane, Amazon RDS poskytuje užívateľom prístup k operačnému systému a umožňuje inštalovať vlastný softvér.
3. Zálohovanie: Azure SQL automaticky zálohovanie databázy a umožňuje obnovu záloh pomocou jednoduchej grafickej konzoly. Amazon RDS poskytuje podobnú funkcionalitu, ale vyžaduje niektoré konfigurácie.
4. Škálovateľnosť: Obidve služby umožňujú horizontálnu i vertikálnu škálovateľnosť, ale Amazon RDS umožňuje aj zmenšenie alebo zvýšenie výkonu pomocou tzv. "scale-up" alebo "scale-down" funkcie.
5. Cena: Ceny pre Azure SQL a Amazon RDS sa líšia podľa typu databázy, veľkosti a počtu používateľov. Všeobecne platí, že Azure SQL je drahší ako Amazon RDS.

V závislosti od vašich potrieb môže byť jeden z týchto poskytovateľov PaaS vhodnejší pre vaše použitie ako druhý.