1)Cos’è un **DBMS**? Qual è il suo scopo? Il DBMS (DataBase Managment System) è un sistema di software ideato per la gestione e l’organizzazione dei dati. Esso consente agli utenti la creazione, la modifica e l’inserimento di query nel database. I due principali tipi di DBMS sono il sistema di gestione di database relazionale(RDBMS), basato sul modello relazionale (dati rappresentati da una raccolta di tabelle messe in relazione tra loro) e il sistema di gestione orientato agli oggetti (OODBMS)(in cui i dati sono rappresentati come oggetti)

2)Indica in maniera dettagliata quali sono le similitudini e le differenze tra **Database Relazionale e Database Orientato agli Oggetti**. Il modello dei RDBMS si basa sul principio di archiviazione dei dati in tabelle (chiamate relazioni) contenenti ognuna delle quali righe e colonne(rispettivamente rappresentanti di un record univoco e di un campo all’interno del record stesso) mentre il modello OODBMS “memorizzano” i dati in oggetti e sono formati da attributi e metodi. Se un DB relazionale utilizza SQL e le relazioni sono gestite tramite chiavi esterne e join, in un DB a oggetti le query dipendono dal linguaggio utilizzato e le relazioni sono gestite direttamente tramite riferimenti agli oggetti. E’ quindi più comodo utilizzare gli RDB per app che richiedono un uso delle query complesse e una gestione strutturata dei dati. Per le applicazioni che gestiscono strutture e relazioni di dati complesse si adattano di più agli OODB.

3)Cos’è un **CRM**? Qual è il suo scopo? Quali sono i suoi utilizzi all’interno di un’azienda? Il CRM (Costumer Relationship Managment) è la strategia di tutte le interazioni e di tutti i rapporti che una determinata azienda ha con tutti i suoi clienti e/o potenziali clienti mirata al miglioramento del rapporto Azienda/clienti. Esso aiuta a semplificarne i processi ea migliorarne la produttività. Può essere inteso come: -processo: sistema che un’azienda adotta per migliorare e consolidare i rapporti con la clientela; - strategia: filosofia aziendale con la quale si gestiscono i rapporti umani con i clienti; -tecnologia: prodotto tecnologico che spesso si basa sui Cloud, utilizzato per gestire e creare report e analisi sulle interazioni tra azienda e clienti. Nell’architettura aziendale il CRM si può dividere in tre macrotipologie: -operativo, soluzioni metodologiche e tecnologiche per automatizzazione dei processi di business; -analitico, strumenti per migliorare la conoscenza del cliente -direzionale, stabilisce le strategie da attuare con un determinato cliente sulle basi delle informazioni estrapolate da CRM op e il CRM an.

4)Cosa si intende per **Field-Level Security**? Il FLS (Field Level Security), è un controllo della gestione della visibilità di un determinato oggetto o campo di un oggetto in base al profilo di un utente o in base ad un Permission Set.

5)A cosa servono le **Validation Rules**? Le VR (Validation Rules) sono dei “paletti” che impongo all’utente l’inserimento o l’aggiornamento di un determinato record impedendogli di inserire informazioni non corrette.

6)Cosa sono i **Record Type**? I Record Types sono utilizzati per raggruppare molti record di un determinato tipo per quell’oggetto consentendo anche di ottenere layout di pagina, campi e valori personalizzati.

7)Qual è la differenza tra **Ruoli e Profili**?I ruoli differiscono dai profili in quanto essi(i ruoli) consentono di creare una gerarchia di utenti al fine di fare accedere ciascun utente ai dati di loro proprietà o con loro condivisi, mentre i Profili definiscono il modo con il quale gli utenti accedono ai dati e le azioni che essi possono svolgere sui dati stessi. “Profiles do, Roles see”

8)Cosa sono gli **Utenti** in Salesforce? Un utente (User) in SF rappresenta una persona che, dopo essersi iscritta inserendo i propri dati e la propria mail, possiede i diritti di accesso a determinati oggetti e/o determinate applicazioni. Le possibilità di visione, di modifica o di aggiunta dipendono, però, dal profilo e dal ruolo ( o dal permission set) assegnategli dall’Admin.

9)Come si compone una relazione **Many-To-Many**? Una relazione Many-to-Many si crea inserendo un oggetto di giunzione che avrà una relazione Uno a Molti con entrambi gli object da mettere in relazione.

10) Cosa si intende per **Permission Set**? A cosa vengono associati? Per Permission Set si intende un insieme di permessi eccezionali che determina a quali record di un oggetto un utente può accedere e quali operazioni può effettuare su di essi.

11) Cosa si inTende per **Data Visibility**? Come viene modificata? Per data Visibility si intende la visibilità e l’accesso ai dati di una determinata app. Essa può essere impostata tramite ruoli, insiemi di autorizzazioni, permision set o profili.

12) Cosa sono i **Permessi CRED**? I permessi CRED (Create, Read, Edit, Delete) sono quei permessi, che assegnati ai vari utenti, permettono l’accesso alla creazione, alla lettura, alla modifica e/o alla eliminazione degli oggetti o dei record di una determinata app.

13) Cos’è un Roll-up Summary? Sottolinea le differenze principali tra **Roll-up Summary e Formula Field**. Il Roll-up Summary è un campo che effettua dei calcoli di riepilogo (somma, sottrazione, calcolo della media, o conteggio) partendo dai record di un oggetto figlio ai record dell’oggetto padre. Esso è dunque disponibile solamente sul record padre se quest ultimo è legato da una relazione master-detail con un altro record. I Formula Fields, invece, calcolano i valori utilizzando i campi all’interno di un singolo record.

14) Qual è la differenza tra relazioni **Master-Detail** e **relazioni Lookup**? La Master-Detail è una relazione fortemente accoppiata tra gli oggetti (all’eliminazione di un record principale vengono eliminati anche tutti i record secondari associati ad esso). La visibilità e la condivisione dei record figli sono dunque dipendenti dal padre. (Può essere creato un Roll-up summary, si elimina a cascata il figlio/ i figli all’eliminazione del padre e dunque non è possibile che esista un record figlio senza un record padre). Le relazioni Lookup sono invece relazioni debolmente accoppiate (non è necessaria l’esistenza di un record padre per la creazione di un record figlio, il campo di roll-up non può essere creato e il record figlio non si elimina all’eliminazione del padre)

15)**Salesforce Object Query Language** è un linguaggio progettato per l’interrogazione di dati all’interno di Salesforce. Permette di estrapolare dati e informazioni specifiche dagli oggetti e ne semplifica l’acceso ai dati tramite i comandi di base includono SELECT, FROM, WHERE.

16)Le **workflow rules** sono strumenti mirati a migliorare l’efficienza operativa e possono essere ‘triggerate’ da vari eventi quali condizioni particolari preimpostate su oggetti, determinate condizioni, regole di aggiornamento dei campi, invio automatico di email secondo regole pre-impostate ecce cc.

17)**Il suffisso finale \_\_r** si usa in Salesforce nelle relazioni tra custom objects per accedere ai campi associati (padre-figlio)

18)**SOSL** è un linguaggio di ricerca che permette in Salesforce di effettuare più ricerche nei dati di piu objects allo stesso tempo e ritorna risultati multipli.

19)Sono dei limiti e delle restrizioni impostate per ogni singolo utente al fine che ogni singolo utente non possa consumare troppe risorse (essendo SF un ambiente dove tutti i clienti condividono la stessa app/server/infrastruttura). Un esempio può essere il numero limite di profili/ruoli che ogni utente (non abbonato) può creare contemporaneamente.

20)In Apex i **processi di automazione** più comuni sono oltre ai trigger, il Process Builder e il WorkFlow. Il primo è uno strumento che permette agli utenti di creare automazioni basate su regole preimpostate senza dover scrivere codice ( può aggiornare i record e inviare notifiche agli utenti ). Il Flow è invece uno strumento per creare automazioni più complesse che permettono di raccogliere informazioni e di aggiornare/integrare dati.

21)E’ un flusso di lavoro nel quale viene consentita la gestione e l’automatizzazione delle approvazioni delle app in questione. Viene utilizzato quando un record deve essere approvato dopo un’esaminazione di uno o piu utenti (in genere admin)

22)**Il Dashboard**, in Salesforce, è una visualizzazione dei dati in maniera sintetica, che raccoglie più Report(documenti che forniscono un’analisi dei dati) e li rappresenta tramite grafici e tabelle al fine di facilitarne le analisi.

23)**Apex** è un linguaggio sviluppato in Salesforce orientato agli oggetti simile a Java, progettato principalmente per la gestione dei dati e per la personalizzazione delle soluzioni in base alle specifiche esigenze di ogni azienda.

24)Sono strutture dati che consentono di raggruppare e/o gestire più elementi assieme. Ne abbiamo per ora studiate tre principali tipi: Liste, Set e Mappe (più le varie possibilità di innestare liste di map, liste di set, liste di liste, liste di liste di map e cosi via). Esse si differenziano principalmente per queste caratteristiche: le **liste** sono ordinate e consentono duplicati, i **set** non sono ordinati e non ammettono duplicati. Le **mappe** hanno invece associata ad ogni valore una chiave unica (es marco=>1, luigi=>3 ecc ecc).

25)**ID** è un codice generalmente unico e univoco assegnato ad un elemento all’interno di un sistema o istanza che serve a rendere questa ultima unica e riconoscibile e distinguibile dalle altre istanze della stessa classe.

26)Una **variabile statica** è una variabile appartenente a una classe che mantiene il suo valore (nel più delle volte già inizializzata nella classe) uguale tra tutte le istanze.

27)Si dice che Apex sia **DATA FOCUSED** in quanto questo linguaggio è stato pensato per operare in modo specifico con i dati già presenti nella piattaforma(Salesforce) ed è fortemente integrato con il Db stesso.

28)Il comando **continue** fa saltare, al determinarsi della condizione preposta, l’iterazione corrente di un ciclo per passare a quella successiva. Il comando break, invece, al verificarsi della condizione preposta, blocca l’esecuzione uscendo dal ciclo.

29)La **classe** è un blueprint, una sorta di modello, di calco sulle quali caratteristiche vengono poi generate una o più istanze del modello in questione e che definisce proprietà (attributi) e comportamenti (metodi) di un oggetto derivante dalla classe stessa.

30)Per **sObject** (o SalesforceObject) si intende una rappresentazione di un oggetto nel dB di Salesforce. Vengono usati per creare, editare, cancellare o aggiornare i record nel sistema.

31) **LWC Lifecycle Hook** è un insieme di metodi che invocati durante il ciclo di vita di un componente Lightning Web Component (connectedCallback(),disconnectedCallback, renderedCallback(), constructor()).

32) **Shadow DOM** è una tecnica che permette di isolare il DOM di un componente dai suoi elementi esterni

33) Per **Deploy** di un Lightning Component si intende il processo di rendere un componente Lightning disponibile in un ambiente di Salesforce.

34) I principali **decorators** per gli LWC sono:

-@wire: collega una proprietà o un metodo a un flusso di dati proveniente da Apex o da una sorgente esterna.

-@api: rende pubblici una proprietà o un metodo, rendendoli accessibile da altri componenti.

-@track: monitora le modifiche apportate ad una proprietà e ne forza l'esecuzione del componente.

35) **Lightning App Builder** è uno strumento di Salesforce che permette di creare applicazioni personalizzate.

36) Il **Controller lato Client** (effettua operazioni locali) è un file JavaScript che gestisce la logica di interazione con l'utente, mentre il Controller lato Server (ha interazioni con Salesforce lato back end) è un file Apex che gestisce la logica e le operazioni sul server.

37) Per abilitare **l'accesso al metodo di un controller lato server in un LWC**, si utilizza la decorazione @wire o si utilizza una funzione Apex annotata con @AuraEnabled rendendo questa ultima accessibile al client.

38) Per eseguire una funzione prima del rendering di un componente si utilizza il lifecycle hook constructor() o connectedCallback().

39) Lightning Components sono basati su un'architettura Aura, mentre gli **LWC** sono basati su tecnologie web standard (come HTML, CSS e JavaScript). Dunque la principale differenza è che LWC è più performante e sfrutta l'uso di web standards.

40) **Il Salesforce Lightning Design System** è una libreria di componenti di interfaccia utente e di linee guida progettate al fine di creare applicazioni che rispettano le best practices di Design in Salesforce.

41) Gli **attributi di un componente in Salesforce** Lightning sono variabili utilizzate per memorizzare dati all'interno del componente stesso e che possono essere modificati nel codice JS.

42) É una direttiva di markup dei componenti Aura che condiziona la visualizzazione di una sezione di codice. Ha un funzionamento simile  all' if nei linguaggi di programmazione infatti permette di rendere visualizzabile o meno una parte di un componente in base a una condizione booleana espressa nell'aura:if stesso.

43) Gli **Helper** in Salesforce Lightning sono funzioni JavaScript utilizzate per separare la logica di business dalla logica di rendering del componente che rendono possibile scrivere porzioni di codice riutilizzabili che possono essere chiamate sia da altri metodi del componente  sia dal controller .

44). È utilizzata per impostare i parametri da passare ad un controller Apex quando si esegue una chiamata asincrona (definisce i valori dei parametri da inviare al server)

45) È un file o una raccolta di file che vengono caricati nel sistema e resi disponibili per le applicazioni Salesforce.

46)  Un componente "**child**" è un componente che è "figlio" di  un altro componente, detto "parent". In Salesforce Lightning, il componente figlio viene implementato nel componente genitore utilizzando la sintassi di markup <c:childComponent />

1. Cosa è **Salesforce**? Risposta: Salesforce è una piattaforma cloud che fornisce soluzioni CRM (Customer Relationship Management) per gestire e automatizzare le vendite, il marketing, e il servizio clienti.

2. Che cos'è **Apex**? Risposta: Apex è un linguaggio di programmazione proprietario di Salesforce, simile a Java, utilizzato per scrivere logiche personalizzate come trigger, classi e metodi sulla piattaforma.

3. Cos'è **SOQ**L? Risposta: SOQL (Salesforce Object Query Language) è un linguaggio simile a SQL, utilizzato per interrogare i dati memorizzati in Salesforce e recuperare informazioni da oggetti personalizzati o standard.

4. Cos'è un **trigger** in Apex? Risposta: Un trigger in Apex è un codice che viene eseguito automaticamente in risposta a eventi specifici su oggetti Salesforce, come la creazione o l'aggiornamento di un record.

5. Qual è la differenza tra un trigger **"before" e "after**"?Risposta: Un trigger "before" viene eseguito prima che i dati vengano salvati nel database, mentre un trigger "after" viene eseguito dopo che i dati sono stati salvati.

6. Che cosa è una **classe Apex**? Risposta: Una classe Apex è un blocco di codice che può contenere metodi e variabili. Viene utilizzata per incapsulare logiche riutilizzabili.

7. Cos'è **il governor limit** in Salesforce? Risposta: I governor limits sono limiti imposti dalla piattaforma Salesforce per evitare l'eccessivo consumo di risorse, come CPU, memoria o numero di query, per ogni transazione.

8. Qual è la differenza tra una **variabile statica e una variabile di istanza** in Apex? Risposta: Le variabili statiche appartengono alla classe e sono condivise da tutte le istanze, mentre le variabili di istanza appartengono a una singola istanza di una classe.

9. Cos'è una **Custom Object** in Salesforce? Risposta: Un Custom Object è un oggetto personalizzato creato in Salesforce per memorizzare dati specifici dell'azienda, non inclusi negli oggetti standard.

10. Come si definisce un campo personalizzato in Salesforce? Risposta: Un campo personalizzato in Salesforce viene definito in un oggetto personalizzato o standard, tramite la creazione di nuovi campi nei layout e nei record.

11. Cosa sono **i record types** in Salesforce? Risposta: I record types consentono di creare diverse varianti di un oggetto in Salesforce, ognuna con layout, regole di validazione e processi aziendali personalizzati.

12. Come si esegue una **query SOQL**? Risposta: Una query SOQL può essere eseguita nel Developer Console di Salesforce, utilizzando la sintassi SELECT field1, field2 FROM Object WHERE condition.

13. Cosa sono i "**Governor Limits**" in Apex? Risposta: I "Governor Limits" sono limiti imposti da Salesforce per garantire che le risorse siano utilizzate in modo efficiente e per evitare che un'applicazione monopolizzi le risorse della piattaforma.

14. Che cos'è una "**DML operation**"? Risposta: Le operazioni DML (Data Manipulation Language) in Apex permettono di eseguire operazioni come insert, update, delete, undelete su oggetti Salesforce.

15. Cos'è un "trigger" di tipo "before insert"? Risposta: Un trigger "before insert" viene eseguito prima che un record venga effettivamente inserito nel database, consentendo la modifica dei dati prima dell'inserimento.

16. Cosa sono i "**Custom Settings**" in Salesforce? Risposta: I "Custom Settings" sono oggetti speciali in Salesforce che memorizzano i dati di configurazione personalizzati, utilizzabili in vari contesti per ottimizzare l'applicazione.

17. Qual è la differenza tra "SOQL" e "SOSL"? Risposta: SOQL è utilizzato per interrogare singoli oggetti, mentre SOSL è utilizzato per effettuare ricerche su più oggetti contemporaneamente.

18. Cos'è un "Test Class" in Apex? Risposta: Una "Test Class" in Apex è una classe progettata per testare il codice Apex e garantire che funzioni correttamente, utilizzando metodi annotati con @isTest.

19. Qual è la sintassi di un'operazione SOQL di base? Risposta: La sintassi di base di SOQL è: SELECT field1, field2 FROM Object WHERE condition.

21. Cos'è il "batch Apex"? Risposta: Il "batch Apex" permette di eseguire operazioni su grandi volumi di dati in modo asincrono, suddividendo il processo in più fasi per rispettare i governor limits.

22. Cos'è una "**Visualforce Page**"? Risposta: Una "Visualforce Page" è una pagina web personalizzata in Salesforce, che utilizza il markup Visualforce per visualizzare e interagire con i dati su Salesforce.

23. Cos'è un " **Builder**" in Salesforce? Risposta: Il "Schema Builder" è uno strumento visivo di Salesforce che consente di visualizzare e creare oggetti, campi e relazioni tra oggetti in un diagramma.

24. Che cos'è un "**Custom Apex Controller**"? Risposta: Un "Custom Apex Controller" è una classe Apex che funge da controller personalizzato per una Visualforce page, gestendo la logica di business e la visualizzazione dei dati.

25. Cos'è un "**Lookup Relationship**"? Risposta: Un "Lookup Relationship" è un tipo di relazione tra due oggetti in Salesforce in cui un oggetto può fare riferimento ad un altro oggetto, creando un legame "uno a uno".

26. Quali sono i limiti di query SOQL in Salesforce? Risposta: I limiti di SOQL in Salesforce includono un massimo di 100.000 record per query e un massimo di 50.000 record nel risultato di una query in un singolo script.

27. Cos'è la "Apex Exception"? Risposta: La "Apex Exception" è un errore che si verifica durante l'esecuzione di un programma Apex. È utilizzata per gestire situazioni anomale e prevenire il fallimento del processo.

28. Cosa significa "data skew" in Apex? Risposta: "Data skew" si verifica quando un oggetto ha un numero sproporzionato di record collegati, causando problemi nelle operazioni di massa, come i trigger o le operazioni DML.

29. Che cos'è un "**Workflow Rule**" in Salesforce? Risposta: Un "Workflow Rule" è una regola che automatizza i processi aziendali in Salesforce, come l'invio di email o la creazione di attività, in base a determinati criteri.

30. Qual è la funzione di un "Queueable Apex"? Risposta: "Queueable Apex" è una tecnica per eseguire operazioni asincrone, simile al batch Apex, ma con maggiore flessibilità e senza i limiti di un batch.