Titlu : SQL
Ciurea Elena Georgiana
31.07.2024

Cuprins

I.Notiuni teroretice

Informatii acumulate ca urmare a parcurgerii cursului de testare manuala

II. Aspecte practice

Punerea in aplicare a cunostintelor dobandite

I.Informatii acumulate ca urmare a parcurgerii cursului de testare manuala

1.Explicati pe scurt ce sunt cerinetele de business , la ce ne folosesc si cine le creeaza

Cerintele de bussiness este un document creat de un analist de bussiness in care se descrie modul de dezvolatere a livrabilelor pe care le va produce .Cerintele de business sunt create de catre diferite parti interesate in cadrul organizatiei .

Cerintele de business ne folosesc ca si ghid pentru echipa de dezvoltare in ceea ce priveste functionalitatile , performantele si caracteristicile asteptate ale produsului sau servicului .

2.Explicați diferența între un test condition și test case

Test condition este o conditie specifica sau un aspect al functionalitatii unui sistem sau software care trebuie testat pentru a ne asigura ca functioneaza corect .

Test case este un set de actiuni sau proceduri detaliate care trebuie urmate pentru a atesta o anumita functionalitate a sistemului sau a unei aplicatii software .

3.Etapele procesului de testare sunt :

Planificare si analiza -in aceasta etapa , se stabilesc obiectivele , resursele si strategiile de testare .Se identifica cerintele si functionalitatile care trebuie testate si se dezvolat un plan de testare.

I .Informatii acumulate ca urmare a parcurgerii cursului de testare manuala

- implementare - in aceasta etapa se valideaza mediu de test prin intermendiul smoke testing , se prioritizeaza testele si se creaza datele de testare.Se grupeaza testele pe baza obiectivelor lor .
- Executie -in aceasta etapa cazurile de testare sunt executate , rezulatatele sunt raportate in tool-ul in care au fost scrise testele .
- Inchidere - se evalueaza criteriile de iesire pentru a inchide testarea in siguranta. Orice taskuri ramase deschise si buguri sunt reevaluate si ulterior inchise .
- 4. Explicați diferența între retesting și regression testing
- Retesting se refera la retestarea unei functionalitati sau an unei zone afectate specifice a software-ului care a fost modificata sau care a suferit o reparare .Scopul retesingului este de a verifica daca functionalitatea respactiva functioneaza acum corect conform asteptarilor.
- Regresion testing este utiliazat pemtru a ne asigura ca modificarile sau adaugirile noi aduse la software nu au cauzat probleme in alte parti ale aplicatiei care erau anterior functionale .
- Prin regresion testing verificam daca noile modificari sau functionalitati adaugate nu au afectat negativ functionalitati existente sau alte zone ale software-ului.

I .Informatii acumulate ca urmare a parcurgerii cursului de testare manuala

5. Explicați diferența între functional testing și non-functional testing

Testarea functionala se refera la - Ce trebuie sa faca produsul ? Verifica daca produsul isi indeplineste functiile. Testele functionale sunt teste scries pe baza specificatiilor si arata ce trebuie sa faca produsul , reprezentand actiuni facute de catre system.

Testarea non-functionala se refera la Cum trebuie sa se comporte produsul? Verifica atribute care descriu cat de bine isi indeplineste sistemul functiile : reiability , mantabilitate, transferabilitate, eficienta , mentenabilitate , transferabilitate, performanta, recuperare, localizare , conformitate.

6. Explicați diferența între blackbox testing și whitebox testing

Black box testing (testare cutie neagra) este o metoda in care testerul examineaza comportamentul si functionaliatea unui sistem fara sa cunoasca detlaiile interne ale implementarii.La testarea black-box nu avem nevoie sa stim ce se intampla in cod .

Testarea White box testing(Testare Cutie alba) reprezinta testarea pe baza structurii interne a unei componente sau a unui sistem si presuspune cunoasterea codului sursa pe baza caruia actioneaza programul.

I.Informatii acumulate ca urmare a parcurgerii cursului de testare manuala

7. Enumerați tehnicile de testare și grupați-le în funcție de categorie (blackbox, whitebox, experience-based).

-Tehinicile de testare backbox sunt :

-Equivalence partitioning (EP)

-Boundary value analysis (BVA)

-State transition testing (STT)

-Decision Table(DT)

Tehinici de testare white box sunt:

-Statement Coverage

-Decision Coverage

8. Explicați diferența între verification și validation

Verification este un proces de verificare a documentelor , designului , codului si programului pentru a verifica daca software -ul a fost construit conform cerintelor sau nu.Verificarea nu implica executarea codului , verificarea foloseste metode precum recenzii , expuneri , inspectii si verificare la birou.Gaseste erori la inceputul ciclui de dezvoltare si vine inainte de validare.

I .Informatii acumulate ca urmare
a parcurgerii cursului de testare manuala

9. Explicați diferența între positive testing și negative testing și dați câte un exemplu din fiecare

Testarea pozitiva inseamna testarea sistemului cu valori pe care sa le poata procesa .Verifica daca aplicatia software se comporta conform asteptarilor cu intrari pozitive sau nu .

De exemplu : exista o functionalitate intr-o aplicatie care accepta numai litere . Introducerea de litere este acceptata iar orice alte valori in afara de acestea nu ar trebui sa fie acceptate. Pentru a efectua o testare pozitiva , introduceti o litera si verificati daca sistemul accepta .

Testarea negativa inseamna testarea cu valori pe care sistemul nu ar trebui sa le poata procesa in mod normal pentru a ne asigura ca acedste valori sunt intr-adevar respinse si ca nu cauzeaza un crash al sistemului.Spre deosebire de tesatrea pozitiva, care analizeaza ce se intaplma in conditiile de reusita , testrea negativa cauta punctele slabe care ar putea aparea in situatii neasteptate , cum ar fi introducerea de date nevalide sau utilizator care incerca sa acceseze o pagina web inerzisa

I .Informatii acumulate ca urmare a parcurgerii cursului de testare manuala

- De exemplu, daca un utilizator incearca sa introduca o litera intr-un camp numeric, comportamentul corect in acest caz ar fi afisarea mesajului „Tip de date incorect, introduceti un număr”. Scopul testarii negative este de a detecta astfel de situatii si de a preveni blocarea aplicatiilor.

10. Enumerați și explicați pe scurt nivelurile de testare

1.Testarea unitara -un test unitar reprezinta celei mai mici bucati functionale dintr-o aplicatie cum ar fi functii, clase , proceduri , interfete .

Testarea unitara este o modalitate prin care fiecare bucata individuala de cod este testata pentru a verifica daca este pregatita pentru untilizare.

2. Testarea de integrare- se concentraza pe interactiunile dintre componente si sisteme.

Obiectivele testarii de integrare : reducerea riscului, verificarea comportamentelor functionale si non-functionale ale interfetelor si comunicarii intre sisteme/componente in raport cu specificatiile.

Exista doua tipuri de testare de integrare:-integrare intre componente si integrare intre sisteme .

I .Informatii acumulate ca urmare a parcurgerii cursului de testare manuala

3. Testarea de sistem-se refera la comportamentul si capabilitarea sistemului ca un tot unitar, tinand cont de comportamentul end-to-end al functionalitatilor pe care sistemul trebuie sa le execute si de comportamentul non-functional asteptat al acelor taskuri.

Obiectivele testarii de sistem: reducerea riscului , verificarea de comportamentelor functionale si non functionale in raport cu cerintele de business, validarea faptului ca sistemul este complet si functioneaza corect,definirea incederii in calitea sistemului complet .

4. Testrea de acceptanta - se concentreaza pe comportamentul produsilui si verifica felul in care aeasta indeplineste nevoile clientului /utilizatorului.

I .Informatii acumulate ca urmare
a parcurgerii cursului de testare manuala

Obiectivele testarii de acceptanta: definirea increderii in calitatea sistemului ca un tot unitar , validarea faptului ca sistemul este complet si functioneaza asa cum ne astepatam, verificarea comportamentelor functionale si non-functionale in raport cu cerintele de business.

Alpha Testing -reprezinta testarea unei aplicatii atunci cand dezvoltarea este completa sau apropape compppleta .In urma testaeii alpha se pot face cateva schimbari minore daca e necesar.

Beta testing are lor la site-ul clientului . Se va trimite sistemul/ softul la utilizatori, acestia vor instala aplicatia si vor incepe sa o foloseasca in conditii reale. Scopul testarii beta este sa puna aplicatia in mainile unor utiliatori reali, oameni ce nu fac parte din echipa de dezvolatori , pentru a descoperi defectele din perspectiva utlizatorului.

II. Punerea in aplicare a cunostintelor dobandite

Baza de date: Employees

1. Instrucțiuni DDL (cel puțin una dintre CREATE, ALTER, DROP, TRUNCATE) :

-create table Departments (Department_id int primary key, Department_name varchar(30), Location varchar(50));

-alter table Departments modify column Location varchar(60);

1. Instrucțiuni de DML (INSERT, DELETE, UPDATE):

-insert into Departments (Department_id, Department_name ,Location) values (124, 'Human resources', 'Bucharest');

-update Employees set hourly_pay = '15'

where Employee_id = 1;

-Delete table Departments;

II.Punerea in aplicare a cunostintelor dobandite
Instrțiuni DQL (select all, select câteva coloane, filtrare cu where, filtrări cu like, filtrări cu AND și OR, funcții agregate, filtrări pe funcții agregate, joinuri - inner join, left join, right join, cross join, limite, order by, chei primare, chei secundare)

- select * from Employees;
- select Salary , Department_id from Details;
- -select * from Location

  where City like 'Bucharest';
- -select* from Details

  where Salary >= '543';
- -select* from employees

  where First_name like 'Ion' and Hourly_pay = 14.2 or Age = 35 ;
- -select location, count(*) from Departments group by location order by count(*) asc;
- -select count(*) from employees;
- select Hourly_pay , avg(Hourly_pay) from Employees group by Hourly_pay having avg(Hourly_pay) <=15;

II. Punerea in aplicare a cunostintelor dobandite

- select Department_id, count(*) from Employees

where Department_id like '12%' group by Department_id

order by count(*) asc;

- select Project_ID, Department_id, count(*) from Projects where Project_ID like '4'

group by Project_ID,Department_id having count(*) <4

order by count(*) asc;

- -select count(*) as 'Employee_id' from employees

where Age >=35;

- select* from Employees

limit 3;

- select* from Projects order by Manager_name desc;

II. Punerea in aplicare a cunostintelor dobandite

- SELECT p.* FROM Projects p where p.Manager_name not in ('Mircea', 'Claudiu');
- select count(*) as 'Location_id' from Location

  where Employee_id =2 and City ='Cluj';
- select* from Departments

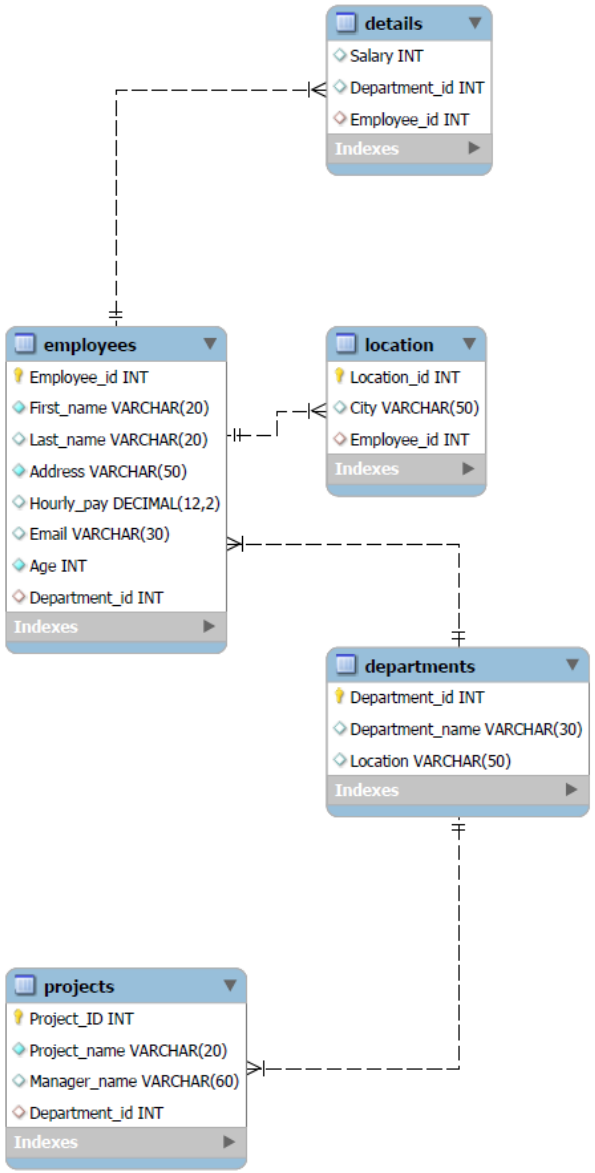
inner join Employees

on Departments.Department_id=Employees.Department_id;
- select* from Departments left join Employees

on Departments.Department_id=Employees.Department_id;
- select* from Departments

right join Employees

on Departments.Department_id=Employees.Department_id;



- select* from Departments inner join Projects on Departments.Department_id= Projects.Department_id;
- select Employee_id from Employees e inner join departments d on e.department_id = d.department_id;
- select department_name from departments inner join Employees on departments.department_id= Employees.department_id;
- select e.employee_id, e.first_name, e.age from Employees as e;

VA MULTUMESC PENTRU ATENTIE !