

→Introduzione:

L'argomento del mio progetto e' uno studio di danza. Ho scelto questo argomento/tema per rappresentarlo in una database perche' mi sembra molto interessante nel contesto dell'organizzazione interna delle attivita'.

In questo progetto ho incluso delle informazioni per quanto riguarda l'organizzazione delle studi di danza e le relazioni(relationships) tra le diverse attivita' incluse. Le attività principali che include sono l'organizzazione dei concerti in luoghi diversi che contribuiscono al guadagno dell'esperienza nella danza ed aiutare presentare e diffondere la cultura.

- *Scopo:*

Uno dei motivi principali del mio progetto e' di analizzare l'organizzazione interna di uno studio di questo genere e di motivare la gente di seguire I loro hobby.

-*Obiettivi:*

- Analizzare il sistema di organizzazione
- Spiegare le funzione e le procedure che si usano in questo database
- Analizzare e spiegare le relationships tra le tabelle contenenti I dati
- Spiegare alle persone che sono interessate e a quelli che non sono interessate, informazione su uno studio di danza

→Descrizione dell'organizzazione:

- *Descrivere l'organizzazione:*

Uno studio di danza e' un attivita molto grande in cui sono incluse quasi tutte le categorie di danza e delle attivita che vuoi fare. Per ogni categoria diversa di danza, lo studio dispone diversi istruttori professionisti che sono dedicati a cio' che insegnano. Un aspetto molto importante e' che tutte le eta' possono trovare se stesso e praticare il loro passione per l'arte della danza. Ci sono diverse classi di danza divise, in base al numero di studenti e alla loro età.

Uno studio di danza e suddivisa in dipartamenti.

- Descrizione del lavoro del progetto:

- La durata del svolgimento di questo progetto è stato circa 3 settimane.

Nella prima settimana ho fatto l'introduzione dello studio e la descrizione dell'organizzazione interna. Ho creato il database DanceStudio ed ho creato le tabelle della database e gli ho popolato:

- TABLE REGION;
- TABLE STUDIO;
- TABLE INSTRUCTOR;
- TABLE DANCE;
- TABLE CONCERT;
- TABLE COMPETE;
- TABLE LOCATEDIN;
- TABLE CLIENT;
- TABLE DANCECLASS;

Nella seconda settimana ho scelto l'elenco delle interrogazioni per supportare il processo decisionale ed ho inserito delle procedure e funzioni. Ci è voluto tutta la settimana perché dovevo scegliere attentamente ciò di cui avevo bisogno per il mio progetto.

Nell'ultima settimana ho fatto lo schema Entita-Relazione e lo Schema delle Relazioni ed ho descritto le entità con gli attributi corrispondenti e le relazioni tra le entità.

Uno studio di danza e suddivisa in dipartamenti:

- *Dipartimento di zumba.* L'istruttore corrispondente insegna la danza e contemporaneamente allena i partecipanti.
- *Dipartimento di hip-hop.* L'istruttore corrispondente insegna una coreografia associata ad una canzone appartenente al gruppo dei canzoni hip hop al gruppo dei partecipanti nel questo dipartimento.

- *Dipartimento di folk dance.* L'istruttore corrispondente al questo dipartimento insegna danze tradizionali provenienti da tutte le regioni.
- *Dipartimento di balletto.* Corrisponde a uno dei dipartimenti piu' difficili e insegna a tutti I giovani il dance classico. E' il dipartimento piu' affollato di tutti.
- *Dipartimento di marketing.* Pubblicizza tutte le attivita' e lo studio stesso ai suoi clienti.
- *Dipartimento di reclutamento.* Si occupa di trovare nuovi ballerini professionisti per rappresentare lo studio di danza.

- Processi / operazioni supportati dal database proposto:

1. Per prima i clienti interessati vengono nel studio e dimostriamo tutti i tipi di attività che offriamo.
2. I clienti scelgono la classe che preferiscono o in cui si trovano di più.
3. Dopo aver scelto la classe si registrano nel nostro studio fornendo il loro nome, cognome, e-mail, numero di telefono e, naturalmente, l'ID personale.
4. Il pagamento dipende per diverse classi e viene eseguita immediatamente dopo la registrazione del cliente.
5. Poi il cliente viene registrato nel database.
6. A causa dell'orario di ogni lezione, i clienti possono partecipare a diverse lezioni di ballo.
7. Se i clienti migliorano le loro abilità di ballo diventano professionisti.
8. I professionisti partecipano ad un gruppo di ballo professionale, che rappresentano lo studio nei concerti.
9. Per ogni concerto e ogni prezzo vinto il database registra tutte le informazioni.
10. Ogni classe, ogni istruttore, ogni cliente ha il suo ID personale.

- Le interrogazioni usate nel database: (7-11 of them)

1. Il manager e' interessato sapere: ClientFirstName, ClientLastName, ClientID, ClientAge e ClientphoneNumber che sono registrati nel folk dance. Ordered by clientID.
2. Il manager e' interessato sapere: StudioName, StudioAddress, StudioID, StudioEmail, ConcertName dello studio che partecipa nei concerti di hip-hop. Ordered by StudioID.
3. Il manager e' interessato sapere: InstructorFirstName, InstructorLastName, InstructorPhoneNumber, InstructorID dei istruttori, ClassName, ClassID e ClassPrice, ClientAge ClientFirstName e ClientLastName where ClientAge<21. Group by

4. Il manager e' interessato sapere: DanceChoreography, DanceMusic, DanceName, DanceType e ClientID della danza che ha finito la coreografia.

5. Il manager e' interessato sapere: DanceChoreography, DanceMusic, DanceName, DanceType, ClientID e ClientLastName dei clienti che la loro DanceType e' il balletto.

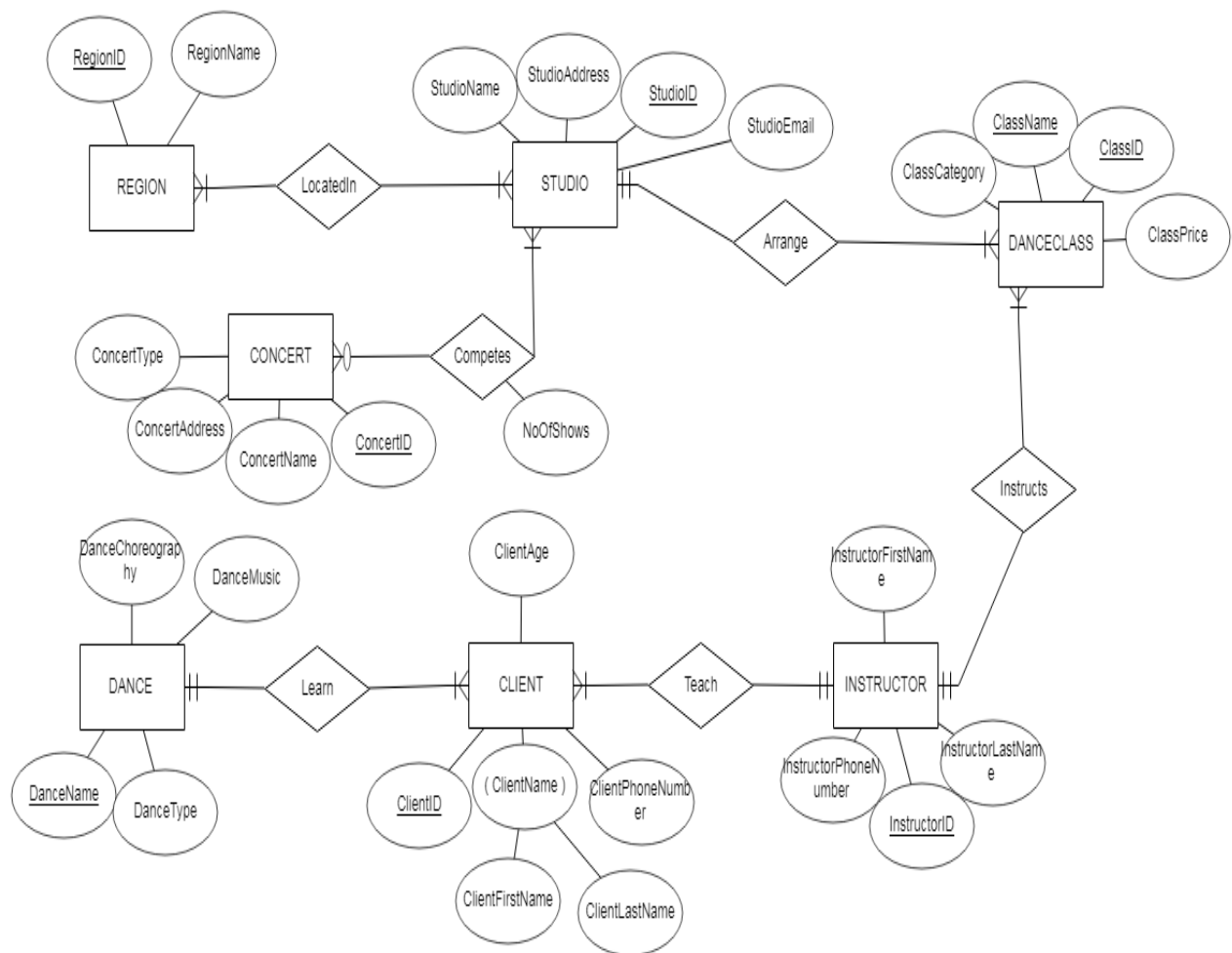
6. Il manager e' interessato sapere: ConcertType, ConcertName, ConcertID, StudioName, StudioID, SUM(NoOfShows) del concerto where il concerto e' un concerto Folk.

7. Il manager e' interessato sapere: ClassID, ClassName, ClassPrice, InstructorFirstName, InstructorID where the ClassPrice<=AVG. Group by ClassID where the ClassCategory='amateur'.

8. Il manager e' interessato sapere: ConcertType, ConcertName, ConcertAddress, ConcertID, NoOfShows of the concerts grouped by ConcertID and other where the NoOfShows is >2.

→ERD

-Diagrama usando ERDPLUS come imagine



-Descrizione dello schema Entita'-Relazione:-

- Le entita' e le loro corrispondenti attributi (E):

REGION(E): RegionName, RegionID(A)

La regione è il luogo in cui si trova lo studio. Ha due attributi che sono: RegionName, che rappresenta il nome della regione in cui si trova lo studio, e RegionID che è l'attributo specifico e fornisce il numero dell'area.

* PK RegionID

STUDIO (E): StudioName, StudioAddress, StudioID, StudioEmail(A)

L'entità Studio rappresenta il luogo in cui si organizzano tutte le attività. Ha quattro attributi che sono: StudioName che rappresenta il nome dello studio, StudioAddress che mostra l'indirizzo

reale del luogo, StudioID che è un attributo specifico e fornisce il numero di studio e StudioEmail che è l'email utile del studio dove tutti possono contattare o ottenere informazioni.

*PK StudioID

*FK RegionID

CLASS(E): ClassPrice, ClassID, ClassName, ClassCategory(A)

L'entità Class ha i suoi attributi che sono: ClassID l'attributo univoco che mostra il numero ID della classe, ClassPrice che informa tutti quanto costa ogni classe di danza, ClassName che rappresenta tutti i diversi tipi di lezioni di danza e quindi tutti possono scegliere, ClassCategory mostra il livello dei clienti che sono: principianti, amatori, professionisti.

*PK ClassID

*FK StudioID, InstructorID

CONCERT(E): ConcertID, ConcertName, ConcertType, ConcertAddress(A)

L'entità Concert ha l'attributo univoco ConcertID che fornisce l'ID no del concerto, ConcertName che rappresenta il nome del concerto, ConcertType l'attributo che ci fornisce informazioni sul tipo di concerto che sarà organizzato(ad esempio un concerto folk, concerto hip-hop ecc.) e l'attributo ConcertAddress che specifica l'indirizzo in cui si terrà il concerto

*PK ConcertID

INSTRUCTOR(E): InstructorFirstName, InstructorLastName, InstructorPhoneNo, InstructorID(A)

L'entità Istruttore ha l'attributo univoco che è InstructorID e mostra il numero personale dell'istruttore, gli attributi InstructorFirstName, InstructorLastName rappresentano il nome e il cognome dell'istruttore della classe e l'attributo InstructorPhoneNo ci fornisce il numero di telefono personale per contattare il istruttore ogni volta che è necessario.

*PK InstructorID

CLIENT(E): ClientID, ClientName, ClientPhoneNo, ClientAge(A)

L'entità Client ha quattro attributi che sono: *ClientID* ha l'attributo univoco che rappresenta il numero ID personale che il client ha, *ClientName* l'attributo che mostra il nome reale del client e *ClientPhoneNo* che indica il numero di ciascun client da utilizzare ogni volta è necessario, ClientAge ci dice l'età di ogni cliente.

*PK ClientID

*FK InstructorID, DanceName

DANCE(E): DanceName, DanceType, DanceChoreography, DanceMusic(A)

L'entità Dance ha solo due attributi: *DanceName* che è univoco perché specifica il nome della danza e l'attributo *DanceType* che separa tutte le danze in diverse categorie, in cui appartengono, *DanceChoreography* che mostra due fasi della coreografia del ballo: finiti oppure no, *DanceMusic* l'attributo che ha tutti i tipi necessari di musica.

*PK DanceName

COMPETE

*PK StudioID, ConcertID, NoOfShows

*FK StudioID, ConcertID

-Descrizione delle Relationships:

- Relazione tra Region&Studio: **LocatedIn**

Questa relazione lega queste due entità. Da entrambe le parti è obbligatorio perché in ogni regione dovrebbe esserci uno studio e ogni studio dovrebbe trovarsi in una regione. La relazione è *molti a molti* perché molti studi possono essere situati in molte regioni e molte regioni possono avere molti studi situati.

- Relazione tra Studio&Concert : **Compete**

Questa relazione lega queste due entità. Il rapporto è opzionale per lo studio perché uno studio non è obbligato a partecipare ad un concerto, mentre la relazione per il concerto è obbligatorio perché ogni concerto dovrebbe avere studi che partecipano a questi concerti. La relazione è *molti a molti* perché molti studi possono prendere parte a molti concerti.

- Relazione tra Studio&Class: **Arrange**

Questa relazione lega queste due entità. Da entrambe le parti la relationship è obbligatorio perché dovrebbe sempre esserci una classe ordinata per uno studio e uno studio che ordina le classi.

Questa relazione è *uno a molti* perché in uno studio potrebbero esserci molte classi e dall'altra parte dovrebbe sempre esserci almeno uno studio che ordina tante classi.

- Relazione tra Class&Instructor: **Instructs**

Questa relazione lega queste due entità. Questa relazione è obbligatoria per entrambe le parti perché dovrebbe sempre esserci un istruttore per allenare una classe e una classe dovrebbe essere sempre allenata dall'istruttore.

La relazione è *uno a molti* perché una classe può avere molti istruttori e un istruttore dovrebbe istruire almeno una classe.

- Relazione tra Client&Instructor: **Teach**

Questa relazione lega queste due entità. È obbligatorio da entrambe le parti perché dovrebbe sempre esserci un istruttore per insegnare le danze ai clienti e dovrebbero essere clienti a cui vengono insegnati le danze.

La relazione è *uno a molti* perché un cliente può essere insegnato/allenato da molti istruttori e un istruttore dovrebbe insegnare/allenare almeno un cliente.

- Relazione tra Client&Dance: **Learn**

Questa relazione lega queste due entità. È obbligatorio in entrambe le parti perché dovrebbe sempre esserci un cliente per imparare una danza e dovrebbe sempre essere una danza che verrà imparata dal cliente.

La relazione è *uno a molti* perché un cliente può imparare tante danze e una danza può essere imparata da almeno un cliente.

-Descrizione delle procedure e funzioni:

1. DanceDescription (Procedura):

Questa procedura prende in ingresso una categoria di danza **'dt'** e restituisce la descrizione delle danze corrispondenti. L'istruzione **CALL**

DanceDescription('dt') si usa per richiamare la procedura con parametro Folk Dance.

2. DanceClassClient (Procedura):

Questa procedura prende in ingresso il nome e il cognome di un cliente e come parametro di uscita il numero totale dei spettacoli. Restituisce il numero totale dei spettacoli fatti dal cliente che verrà inserito nell'istruzione **CALL**

DanceClassClient('ClientFirstName', 'ClientLastName', 'NoShows').

3. InsertClient (Procedura):

Questa procedura permetta di inserire nuovi clienti nella tabella **CLIENT**. Nella procedura è definito un handler di tipo **EXIT** per l'errore **SQLSTATE '23000'** che gestisce la violazione

del vincolo di integrità referenziale tra la tabella **CLIENT** e le tabelle **INSTRUCTOR** e **DANCE**.

-SQLSTATE '23000' : can not add or update a child row: a foreign key constraint fails.

4. ViewClient (Vista):

Questa vista filtra l'inserimento di un cliente nella tabella CLIENT in modo che sia possibile inserire un cliente di eta' maggiore a 8, oppure minore o uguale a 20.

5. NumberOfCompetitions (Funzione):

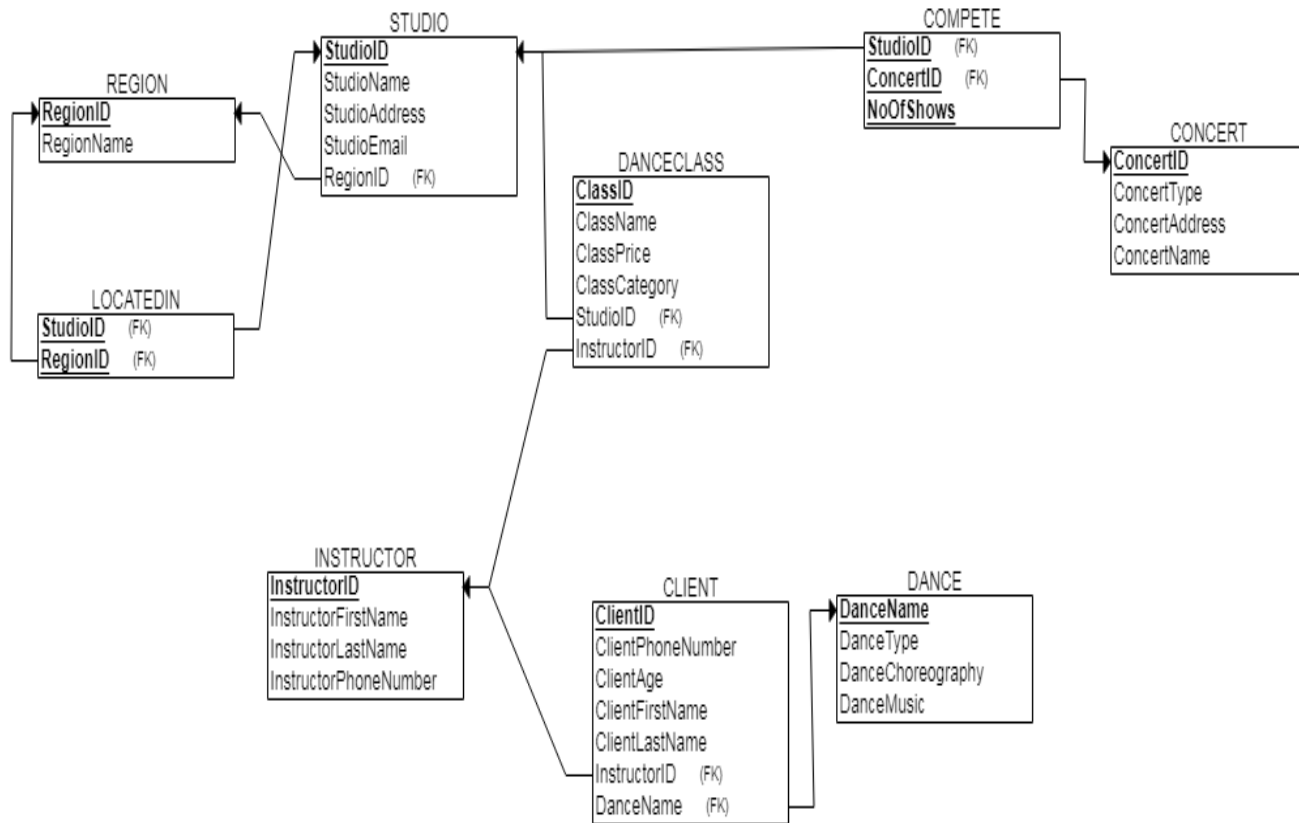
Questa funzione prende in ingresso ID dello studio, ID del concerto e una variabile 'y' che rappresenta un'anno. La funzione restituisce il numero delle comeptizioni fatte da uno studio in quell'anno 'y'.

6. UpdateRegion (Trigger):

Ho creato un trigger chiamato UpdateRegion che viene invocato subito prima (*BEFORE*) dell'aggiornamento (*UPDATE*) della tabella REGION. All'interno del blocco BEGIN-END, effettuiamo un inserimento dei dati all'interno della tabella *Region_log* dove registriamo i valori vecchio e nuovo per i campi "RegionName" ed "RegionID" della tabella REGION. Per effettuare quest'operazione utilizziamo le keywords **OLD** e **NEW** per far riferimento, rispettivamente, al valore precedentemente alla modifica ed a quello nuovo, successive all'operazione di UPDTE.

→ Schema Relazionale

-Usato ERDplus



→ Riassunto

Questa è la fine del mio progetto. Come menzionato prima include tutte le parti importanti di uno "Dance Studio" come: l'analisi dell'organizzazione, la rappresentazione, gli obiettivi, le interrogazioni, il diagramma ERD, lo schema relazionale ERD, le istruzioni SQL.

Anche se non è stato un progetto facile, mi è davvero piaciuto farlo e lavorarci su. Ha inoltre rafforzato le mie conoscenze nel Basi di Dati e Sistemi Informativi. Questo database ci dà la possibilità di vedere in un modo diverso questo tipo di attività che ha una parte importante nella nostra vita quotidiana. Questo è solo un aspetto del questo business, anche se ho cercato di mettere tutte le informazioni necessarie.

Inoltre, alcune delle informazioni menzionate in precedenza sono presentate nel file di testo in cui tutti i codici SQL sono scritti per un database di una danza da studio.