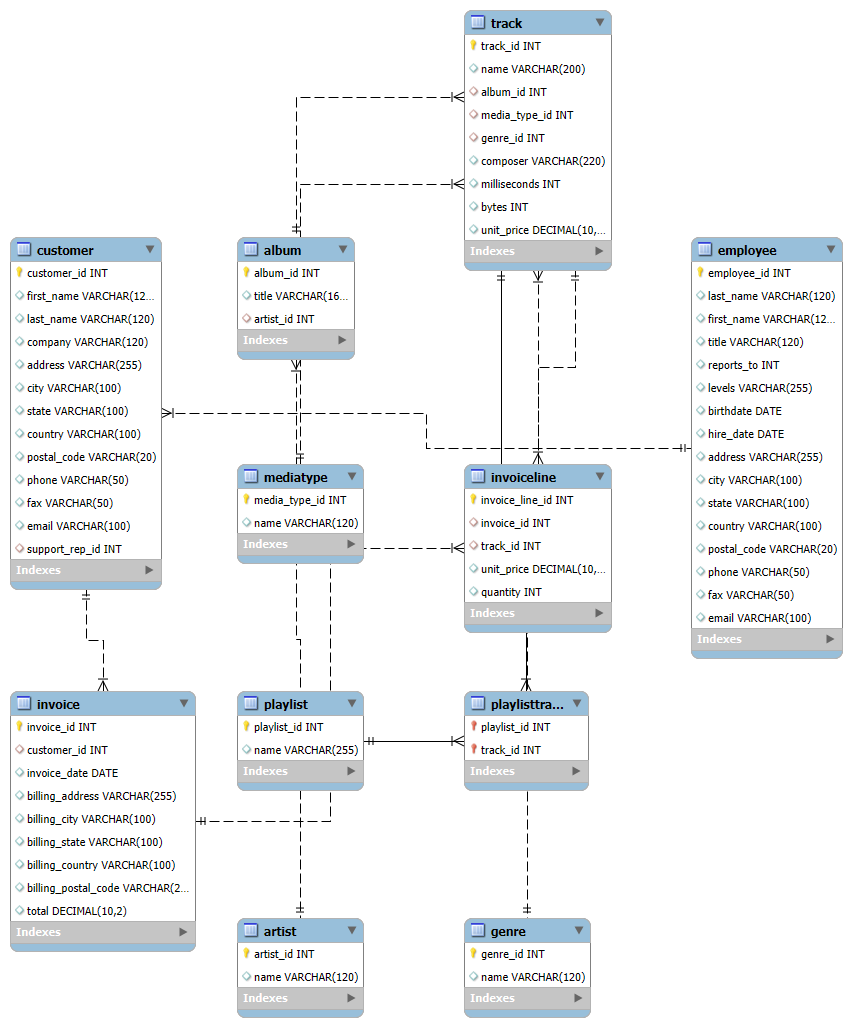
**ERR Diagram:**



**1. Customer Table**

**Primary Key (PK):** customer\_id

**Attributes:** first\_name, last\_name, company, address, city, state, country, postal\_code, phone, fax, email, support\_rep\_id.

**Foreign Key (FK):** support\_rep\_id → employee.employee\_id

(Each customer may have a support representative assigned.)

**Relationships:**

One-to-Many: Customer → Invoice (a customer can have many invoices).

Many-to-One: Customer → Employee (many customers can be assigned to one employee).

**2. Employee Table**

**PK:** employee\_id

**Attributes:** last\_name, first\_name, title, reports\_to, levels, birthdate, hire\_date, address, city, state, country, postal\_code, phone, fax, email.

**FK:** reports\_to → employee.employee\_id (self-referencing hierarchy).

**Relationships:**

One-to-Many (self): An employee can manage other employees.

One-to-Many: Employee → Customer (employees support many customers).

**3. Invoice Table**

**PK:** invoice\_id

**Attributes:** customer\_id, invoice\_date, billing\_address, billing\_city, billing\_state, billing\_country, billing\_postal\_code, total.

**FK:** customer\_id → customer.customer\_id

**Relationships:**

Many-to-One: Invoice → Customer (an invoice belongs to a single customer).

One-to-Many: Invoice → InvoiceLine (an invoice contains multiple lines/tracks).

**4. InvoiceLine Table**

**PK:** invoice\_line\_id

**Attributes:** invoice\_id, track\_id, unit\_price, quantity.

**FKs:**

invoice\_id → invoice.invoice\_id

track\_id → track.track\_id

**Relationships:**

**Many-to-One:** InvoiceLine → Invoice (line belongs to one invoice).

**Many-to-One:** InvoiceLine → Track (line references one track).

**5. Track Table**

**PK:** track\_id

**Attributes:** name, album\_id, media\_type\_id, genre\_id, composer, milliseconds, bytes, unit\_price.

**FKs:**

album\_id → album.album\_id

media\_type\_id → mediatype.media\_type\_id

genre\_id → genre.genre\_id

**Relationships:**

Many-to-One: Track → Album, Track → Mediatype, Track → Genre.

One-to-Many: Track → InvoiceLine (a track can appear on many invoices).

Many-to-Many: Track ↔ Playlist (via PlaylistTrack).

**6. Album Table**

**PK:** album\_id

**Attributes:** title, artist\_id.

**FK:** artist\_id → artist.artist\_id

**Relationships:**

Many-to-One: Album → Artist.

One-to-Many: Album → Track (an album contains multiple tracks).

**7. Artist Table**

**PK:** artist\_id

**Attributes:** name.

**Relationships:**

One-to-Many: Artist → Album (an artist can have multiple albums).

**8. MediaType Table**

**PK:** media\_type\_id

**Attributes:** name.

**Relationships:**

One-to-Many: Mediatype → Track (tracks are available in various media types).

**9. Genre Table**

**PK:** genre\_id

**Attributes:** name.

**Relationships:**

One-to-Many: Genre → Track.

**10. Playlist Table**

**PK:** playlist\_id

**Attributes:** name.

**Relationships:**

Many-to-Many: Playlist ↔ Track (via PlaylistTrack).

**11. PlaylistTrack Table**

**Composite PK:** (playlist\_id, track\_id)

**FKs:**

playlist\_id → playlist.playlist\_id

track\_id → track.track\_id

**Relationships:**

Implements many-to-many relationship between Playlist and Track.

**Relationships Overview:**

**One-to-Many:**

Customer → Invoice

Invoice → InvoiceLine

Employee → Customer

Employee (self) → Employee (reports\_to hierarchy)

Artist → Album

Album → Track

Genre → Track

MediaType → Track

Track → InvoiceLine

**Many-to-Many:**

Playlist ↔ Track (via PlaylistTrack).

**One-to-One:**

Not explicitly present (though Employee self-reporting can mimic hierarchy but not strictly one-to-one).

Not explicitly present (though Employee self-reporting can mimic hierarchy but not strictly one-to-one).