அலகு IV

பாடம் 12

வினவல் அமைப்பு மொழி (SQL)

QFV3UM

கற்றலின் நோக்கங்கள்

இந்தப் பாடப்பகுதியைக் கற்றபின் மாணவர் அறிந்து கொள்வது:

- SQL வினவல் மொழியின் செயலாக்க திறன்.
- SQL-ன் கூறுகள்.
- புலங்கள் மற்றும் பதிவுகளை குறிப்பிட்டு ஒரு அட்டவணையை உருவாக்குதல்.
- ஒரு அட்டவணையில் பதிவுகளை சேர்த்தல், மேம்படுத்துதல் (புதுப்பித்தல்) மற்றும் நீக்குதல் போன்ற பல்வேறு கையாளுதலை செயற்படுத்துதல்.
- பல்வேறு நிபந்தனைகளையும் அவற்றை அட்டவணைகளில் பயன்படுத்தும் வழிமுறைகளையும் தெரிந்து கொள்ளுதல்.
- பல்வேறு உட்பிரிவுகளைப் பயன்படுத்துவதன் மூலம் அட்டவணையில் வினவல்களை உருவாக்குதல்.
- ஏற்கனவே உள்ள அட்டவணையின் கட்டமைப்பை மாற்றியமைக்க.
- ஒரு அட்டவணை மற்றும் பதிவுகளை நீக்குவதற்கான கட்டளைகளையும், செயல்படுத்தப்பட்ட கட்டளைகளை திரும்ப பெறும் கட்டளைகள் பற்றி அறிந்து கொள்ளுதல்.

12.1 SQL அறிமுகம்

வினவல் அமைப்பு மொழி (Structured Query Language) என்பது தரவு தளங்களை அணுகுதலுக்கும், கையாளுதலுக்கும் பயன்படுத்தப்படும் ஒரு வினவல் மொழியாகும். SQL பயனருக்கு தரவுதளங்களை உருவாக்குதல், திரும்ப எடுத்தல், மாற்றி அமைத்தல் மற்றும் தகவல்களை பரிமாற்றம் செய்தல் போன்ற செயற்பாடுகளை செயற்படுத்த அனுமதிக்கின்றது. உறவுநிலை தரவுதள மேலாண்மை அமைப்பில் (Relational DataBase Management System) உள்ள தரவுகளை நிர்வகிப்பதற்கும், அணுகுவதற்கும் வடிவமைக்கப்பட்ட ஒரு மொழியாகும்.

SQL பல பதிப்புகள் கொண்டது. SQL-ன் அசல் பதிப்பு IBMன் ஆய்வு மையத்தில் உருவாக்கப்பட்டது. மேலும், இது 1970 தொடக்கத்தில் Sequel என்று அழைக்கப்பட்டது. பின்னர், SQL என மாற்றப்பட்டது. American National Standard Institute (அமெரிக்க தேசிய தரநிலை நிறுவனம்) நிறுவனத்தால் 1986-ல் வெளியிடப்பட்ட SQL தரநிலை, 1992ல் புதுப்பிக்கப்பட்டு, SQLன் அண்மை பதிப்பு 2008 ல் வெளியிடப்பட்டு, SQL 2008 என பெயரிடப்பட்டுள்ளது.

Chapter 12.indd 208 30-01-2020 20:45:16

குறிப்பு

SQL ன் தற்போதைய அண்மைகால பதிப்பு, 2008 ஆம் ஆண்டு வெளியிடப்பட்டுள்ளது.

12.2 உறவுநிலை தரவுதள மேலாண்மை அமைப்பில் SQL -ன் பங்கு

RDBMS என்பதன் விரிவாக்கம் Relational DataBase Management System என்பதாகும். Oracle, MySQL, MS SQL Server, IBM DB2 மற்றும் மைக்ரோ சாப்ட் Access போன்றவை உறவுநிலை தரவுதள மேலாண்மை அமைப்பதற்கான RDBMS மென்பொருள் தொகுப்புகளாகும். அத்தகைய தரவுத்தளங்களுடன் தொடர்புக் கொண்டு தரவுகளை கையாள SQL மொழி பயன்படுகின்றது.

பொதுவாக, **தரவுத்தளம்** என்பது தொடர்புடைய தரவுகளை கொண்ட அட்டவணையின் தொகுப்பைக் கொண்ட ஒரு களஞ்சியம் எனப்படும். பிற பயன்பாடுகள் பயன்படுத்தும் வகையில் இவற்றை வினவல்கள் கொண்டு செயற்படுத்தி கொள்ளலாம். தரவுத்தள மேலாண்மை அமைப்பு தரவுதளத்தை உருவாக்குதல், நிர்வகிக்கும் திறன் மற்றும் தரவுத்தள பணித்தளத்தினை பயன்படுத்தும் திறன் போன்றவற்றை ஆதரிக்கிறது. RDBMS என்பது தொடர்புடைய தரவு கூறுகளை இணைக்கும் வரிசை அடிப்படையிலான அட்டவணை வடிவமைப்பைக் கொண்ட ஒரு வகை தரவுத்தள மேலாண்மை அமைப்பாகும். மேலும், Create, Read, Update மற்றும் Delete செயற்பாடுகளை ஒட்டுமொத்தமாக CRUD எனப்படும்.

RDBMS-ல் உள்ள தரவுகள், தரவுத்தள பொருட்களாக **அட்டவணைகளில்** (Tables) சேமிக்கப்படும். ஒரு அட்டவணை என்பது தொடர்புடைய தரவுப் பதிவுகளின் தொகுப்பாகும், அட்டவணை வரிசைகள் மற்றும் நெடுவரிசைகளை கொண்டிருக்கும்.

ஒரு அட்டவணையின் நெடுவரிசை **புலம்** (field) என்று குறிப்பிடப்படுகிறது, இது ஒவ்வொரு பதிவின் தேவையான தகவல்களை பராமரிக்கும் வகையில் வடிவமைக்கப்பட்டுள்ளது. ஒரு அட்டவணையின் நெடுவரிசையை ஒரு குறிப்பிட்ட புலத்தின் தொடர்புடைய அனைத்து தகவல்களையும் கொண்டிருக்கும். ஒரு மாணவர் அட்டவணையில் புலங்கள் AdmNo, studName, studAge, studclass, place ஆகிய புலங்களைக் கொண்டிருக்கலாம்.

ஒரு அட்டவணையில் உள்ள தொடர்புடைய புலங்கள் அல்லது நெடுவரிசைகளின் தொகுப்பைக் கொண்ட ஒரு வரிசை, **பதிவு** (Record) எனப்படும். ஒரு பதிவு என்பது மாணவர் அட்டவணையில் உள்ள ஒரு குறிப்பிட்ட மாணவரின் விவரங்களை கொண்டிருக்கும் ஒரு வரிசையாகும்.

12.3 SQL-ன் செயலாக்க திறன்கள்

SQLன் பல்வேறு செயலாக்க திறன்கள் :

1. **தரவு வரையறை மொழி** (DDL-**Data Definiton Language**) : SQL DDL-அட்டவணைகளை (Relational) உறவுநிலை திட்டங்களை (கட்டமைப்பு) வரையறுத்தல், நீக்குதல், மாற்றியமைத்தல் மற்றும் குறிப்புகளை உருவாக்கத்திற்கு தேவையான கட்டளைகளை வழங்கும் ஒரு வரையறை மொழியாகும்.
2. **தரவை கையாளுதல் மொழி (DML-Data Manipulation Language)** : SQL DML தரவுத்தளங்களில் வரிசைகளை (tuples)-ஐ சேர்க்கவும், நீக்கவும், மாற்றவும் கட்டளைகளை உள்ளடக்கியது.
3. **உட்பொதிக்கப்பட்ட தரவுக் கையாளுதல் மொழி** (EDML-**Embedded Data Manipulation Language) : SQLன்** உட்பொதிந்த வடிவம், உயர்நிலை நிரலாக்க மொழிகளில் பயன்படுத்தப்படுகிறது.

Chapter 12.indd 209 30-01-2020 20:45:16

4. **View வரையறை** : SQL அட்டவணையை பார்வையிடுவதற்கான View கட்டளைகளைக் கொண்டுள்ளது.

5. **அங்கீகாரம்** : SQL உறவுநிலைக்கான அணுகல் உரிமைகளுக்கான கட்டளைகளையும், அட்டவணைகளை பார்வையிடுதலுக்கான கட்டளைகளையும் கொண்டுள்ளது.

6. **ஒருமைப்பாடு** : SQL நிபந்தனையை பயன்படுத்தி ஒருமைப்பாட்டை சரிபார்த்தலுக்கு படிவங்களை வழங்குகிறது.

7. **பரிவர்த்தனை கட்டுப்பாடு** : கோப்பு பரிமாற்றங்களுக்கான கட்டளைகள் மற்றும் பரிவர்த்தனை செயலாக்கத்தின் கட்டுப்பாட்டையும் SQL கொண்டுள்ளது.

உங்களுக்குத் தெரியுமா **SQL(Structured Query Language)**-வினவல் அமைப்பு மொழி தரவுத்தளங்களை அணுகுதலுக்கான மொழி எனில், **MySQL** என்பது **SQL Server, Oracle, Informix, Postgres, etc.** போன்று ஒரு தரவுத்தள மேலாண்மை அமைப்பாகும். **MySQL** ஒரு உறவுநிலை தரவுத்தள மேலாண்மை அமைப்பு **(RDBMS).**

MYSQL- நிறுவுதல் இணைப்பு 1-ல் காண்க

12.4 ஒரு தரவுத்தளத்தை உருவாக்குதல்

1. ஒரு தரவுத்தளத்தை உருவாக்க, பின்வரும் கட்டளையை தூண்டுகுறியில் உள்ளிடவும்:

   CREATE DATABASE database_name;

   தரவுத்தளத்தின் பெயர்

   (எடுத்துக்காட்டாக) அட்டவணைகளை சேமித்து வைக்க ஒரு தரவுத்தளத்தை உருவாக்க:

   CREATE DATABASE stud;

2. தரவுத்தளத்துடன் செயலாற்ற, பின்வரும் கட்டளையை உள்ளிடவும்.

   USE DATABASE;

   எடுத்துக்காட்டாக, உருவாக்கப்பட்ட **stud** என்ற தரவுத்தளத்தை பயன்படுத்த கொடுக்க வேண்டிய கட்டளை

   USE stud;

TUJ62X

உங்களுக்குத் தெரியுமா WAMP என்பதன் விரிவாக்கம் "Windows, Apache, MySQL மற்றும் PHP". WAMP எனப்படும் சொல்லானது விண்டோஸ், அப்பாச்சி, MySQL, PHP என்ற சொற்களின் முதல் எழுத்துக்களை எடுத்து உருவாக்கப்பட்ட சுருக்கப் பெயர் ஆகும். இணைய வளர்ச்சிக்கும், உட்புற சோதனைக்கும் இது பயன்படுத்தப்படுகிறது. இணையதள நேரலை சேவைகளுக்கு பயன்படுத்திக் கொள்ளலாம்.

12.5 SQL-ன் கூறுகள்

SQL- **கட்டளைகள் ஐந்து பொதுப்பிரிவுகளாக பிரிக்கப்பட்டுள்ளது:**

Chapter 12.indd 210 30-01-2020 20:45:16

- DML - தரவு கையாளுதல் மொழி
- DDL - தரவு வரையறை மொழி
- DCL - தரவு கட்டுப்பாட்டு மொழி
- TCL - பரிவர்த்தனை கட்டுப்பாட்டு மொழி
- DQL - தரவு வினவல் மொழி

12.5.1 தரவு வரையறை மொழி

தரவு வரையறை மொழி (DDL) தரவுத்தள அமைப்பு அல்லது திட்ட வடிவமைப்பினை வரையறுக்கும் SQL கூற்றுகளை கொண்டிருக்கும். இது தரவுத்தள திட்ட வடிவமைப்பின் விளக்கங்களுடன் செயல்படுகிறது. மேலும், தரவுத்தளங்களில் உள்ள தரவுத்தள உறுப்புகளில் கட்டமைப்பை உருவாக்குவதற்கும், மாற்றியமைப்பதற்கும் பயன்படுகிறது.

DDL தரவுத்தள அமைப்பு சேமிப்பு வடிவமைப்பையும், அணுகல் முறைகளையும் குறிப்பிடப் பயன்படும் வரையறைகளை வழங்குகிறது.

ஒரு DDL கீழ்க்காணும் செயல்பாடுகளை செயல்படுத்துகிறது :

1. தரவு உருப்படி, பிரிவு, பதிவு மற்றும் தரவுத்தள கோப்பு போன்ற தரவு பிரிவு வகைகளை அடையாளம் காண வேண்டும்.
2. ஒவ்வொரு தரவு உருப்படி, பதிவு, கோப்பு மற்றும் தரவுத்தள வகைகளுக்கு ஒரு தனித்தன்மையான பெயரை கொடுக்கிறது.
3. இது சரியான தரவு வகையை குறிப்பிட வேண்டும்.
4. இது தரவு உருப்படியின் அளவை வரையறுக்க வேண்டும்.
5. ஒரு தரவு உருப்படி பயன்படுத்தக்கூடிய மதிப்புகளின் வரம்பை இது வரையறுக்கலாம்.
6. அங்கீகரிக்கப்படாத தரவு பதிவுகளை தடுப்பதற்கான தனியுரிமையை குறிப்பிடலாம்.

தரவு வரையறை மொழியின் கீழ்வரும் SQL கட்டளைகள் பின்வருமாறு:

CREATE	தரவுத்தளத்தில் அட்டவணைகளை உருவாக்க
ALTER	தரவுத்தளத்தின் வடிவமைப்பை மாற்றியமைக்க
DROP	தரவுத்தளத்தில் உள்ள அட்டவணைகளை நீக்க
TRUNCATE	ஒரு அட்டவணையில் உள்ள அனைத்து பதிவுகளையும் அழிக்கும். மேலும், அவற்றுக்கென ஒதுக்கப்பட்ட இடத்தையும் விடுவிக்கும்.

Chapter 12.indd 211 30-01-2020 20:45:16

12.5.2 தரவு கையாளுதல் மொழி

தரவு கையாளுதல் மொழி (DML) என்பது ஒரு தரவுத்தளத்தில் உள்ள தரவுகளை சேர்த்தல், அழித்தல் மற்றும் மாற்றியமைப்பதற்காக பயன்படும் ஒரு கணிப்பொறி நிரலாக்கு மொழியாகும். SQL-ல் தரவு கையாளுதல் மொழி தரவுத்தள அட்டவணையில் சேமித்து வைக்கப்பட்டுள்ள தரவுகளை மாற்றியமைக்க பயன்படும் SQL தரவு மாற்று கூற்றுகளை கொண்டிருக்கும்.

தரவுத்தள திட்ட வடிவமைப்பின்படி, தரவுத்தளம் உருவாக்கிய பின்னர், DML ல் குறிப்பிட்டுள்ள செயல்முறைகளின் தொகுப்பை பயன்படுத்தி தரவுகளை கையாளலாம்.

தரவுகளை கையாளுதல் என்பது,

- தரவுத்தளத்தில் புதிய தகவலை சேர்த்தல்
- தரவுத்தளத்தில் சேமித்து வைக்கப்பட்டுள்ள தகவல்களை திரும்ப பெறுதல்.
- தரவுத்தளத்திலிருந்து வேண்டாத தகவல்களை நீக்குதல்.
- தரவுத்தளத்தில் சேமித்து வைக்கப்பட்டுள்ள தரவுகளை மாற்றி அமைத்தல்.

DML (தரவு கையாளுதல் மொழி) அடிப்படையில் இரண்டு வகைப்படும்:

செயல்முறையுடனான DML – தேவைப்படும் தரவையும், அதனை பெறும் வழிமுறையும் பயனரால் குறிப்பிடப்பட வேண்டும்.

செயல்முறை அல்லாத DML - பயனர் தேவைப்படும் தரவை மட்டும் குறிப்பிட்டால் போதுமானது அதனை பெற தேவையான வழி முறையை குறிப்பிடத் தேவையில்லை.

SQL-லில் உள்ள தரவு கையாளுதல் மொழியின் கட்டளைகள்:

INSERT	ஒரு அட்டவணையில் தரவுகளை நுழைத்தல்.
UPDATE	அட்டவணையில் ஏற்கனவே உள்ள தரவுகளை புதுப்பித்தல்.
DELETE	அட்டவணையில் உள்ள அனைத்து பதிவுகளையும் நீக்கும். ஆனால் அவற்றிற்கு ஒதுக்கப்பட்ட நினைவக பகுதியை விடுவிக்காது.

12.5.3 தரவு கட்டுப்பாட்டு மொழி

ஒரு தரவு கட்டுப்பாடு மொழி என்பது தரவுத்தளத்தில் சேமித்து வைக்கப்பட்டுள்ள தரவுகளின் அணுகுதலை கட்டுப்படுத்தும் ஒரு நிரலாக்க மொழியாகும். இது தரவுத்தளத்தை அணுகுவதற்கான சிறப்புரிமையை கட்டுப்படுத்துகிறது (அங்கீகாரம்). இத்தகைய சிறப்புரிமைகள் வரிசைத்தொடர்களை உருவாக்கல், அட்டவணைகளை பார்வையிடுதல் போன்ற அனைத்து தரவுத்தள செயல்பாடுகளை செயல்படுத்துவதைக் குறிக்கும்.

SQL-லில் உள்ள தரவு கட்டுப்பாட்டு மொழியின் கட்டளைகள்:

GRANT	ஒரு குறிப்பிட்ட பணியை செய்ய ஒன்று அல்லது அதற்கு மேற்பட்ட பயனர்களுக்கு அனுமதி வழங்கும்.
REVOKE	Grant கட்டளையினால் தரப்பட்ட அணுகல் அனுமதி திரும்பப் பெறப்படும்.

Chapter 12.indd 212 30-01-2020 20:45:16

12.5.4 பரிவர்த்தனைக்கான கட்டுப்பாட்டு மொழி

தரவுத்தளத்தில் உள்ள பரிவர்த்தனைகளை (TCL) நிர்வகிக்க பரிவர்த்தனை கட்டுப்பாட்டு மொழி பயன்படுகிறது. DML கூற்றுகளின் மூலம் அட்டவணையில் உள்ள தரவுகளுக்கு செய்யப்படும் மாற்றங்களை நிர்வகிக்க அவை பயன்படுத்தப்படுகின்றன.

SQL – லில் உள்ள பரிவர்த்தனை கட்டுப்பாட்டு மொழியின் கட்டளைகள்:

Commit	தரவுத்தள பரிவர்த்தனையை நிரந்தரமாக சேமிகும்.
Roll back	ஒரு தரவுத்தளத்தை முந்தைய commit நிலைவரை மீட்டெடுக்கும்.
Save point	Rollback செய்வதற்கு ஏதுவாக தரவுத்தள பரிவர்த்தனையை தற்காலிமாக சேமிக்கும்.

12.5.5 தரவு வினவல் மொழி

தரவு வினவல் மொழி ஒரு தரவுத்தளத்தில் உள்ள தரவுகளை வினவுதலுக்கும், மீட்டெடுப்பதற்கான கட்டளைகளை கொண்டுள்ளது. தரவு வினவல் மொழியில் உள்ள அத்தகைய ஒரு SQL கட்டளை SELECT ஆகும்.

Select	அட்டவணையிலுள்ள பதிவுகளை வெளிக்காட்டும்.

12.6 தரவு வகைகள்

தரவுத்தளத்தில் தரவுகள், அவற்றில் இருத்தி வைக்கப்படும் மதிப்புகளின் அடிப்படையில் சேமிக்கப்படும். இது தரவின் தரவு வகையாக அடையாளப் படுத்தப்படுகிறது (அல்லது) ஒவ்வொரு புலமும் ஒரு தரவு வகையைச் சேர்ந்ததாக இருக்கலாம். கொடுக்கப்பட்ட புலத்தில் உள்ள அனைத்து மதிப்புகளையும் ஒரே வகையாக இருக்க வேண்டும்.

SQL ன் ANSI தரநிலை உரை மற்றும் எண் தரவு வகைகளை மட்டுமே அங்கீகரிக்கிறது. ஆனால் சில தனியார் நிரல்கள் தேதி மற்றும் நேரம் போன்ற தரவு வகைகளை பயன்படுத்துகின்றன. ANSI தரவு வகைகள் கீழே அட்டவணை 12.1-ல் பட்டியலிடப்பட்டுள்ளன.

தரவு வகை	விளக்கம்
char (Character)	நிலையான அகலம் கொண்ட சர மதிப்பு. இந்த வகையின் மதிப்பு ஒற்றை மேற்கொள் குறிகளுக்குள் தரப்படும். (எ.கா.) Anu's என்பது 'Anu"S' என்று குறிப்பிடப்படும்.
varchar	மாறும் அகலம் கொண்ட குறியுரு சரம். இது char தரவு வகையை போன்று இருந்தாலும் தரவின் அளவிற்கேற்ப பதியும் அளவு மாறும்.

Chapter 12.indd 213 30-01-2020 20:45:16

dec (Decimal)	15.12,0.123 போன்ற பின்ன எண்களை இது குறிக்கும். இங்கு அளவுக்கான செயலுருபு இரண்டு பகுதிகளை கொண்டுள்ளது. துல்லியம் மற்றும் அளவுகோல். அளவு செயலுருபுவில் வரும் துல்லியம் எத்தனை இலக்க எண் என்பதையும், அளவு கோல் தசமபுள்ளிக்கு வலப்பக்கம் அதிகபட்சமாக எத்தனை இலக்கங்கள் இடம் பெற வேண்டும் என்பதை குறிக்கம். size(5,2) என்பதில் துல்லியம் 5 ஆகவும் மற்றும் அளவுகோல் 2 எனவும் எடுத்துக் கொள்ளும். அளவுக்கோல் துல்லியத்தை விட அதிகமாக இருக்கக் கூடாது.
numeric	இது decimal தரவு வகை போன்றே இருக்கும். ஆனால் இலக்கங்களின் எண்ணிக்கை துல்லிய செயலுருபுவைக் காட்டிலும் அதிகமாக இருத்தல் கூடாது.
int (Integer)	இது தசமபுள்ளி இல்லாத ஒரு முழு எண்ணைக் குறிக்கும். இது size அளபுரு பயன்படுத்தாது.
smallint	இது integer வகையை போன்று காணப்பட்டாலும் கொடாநிலை அளவு integer ஐ விட சிறியதாக இருக்கும்.
float	இது பதின்ம எண் அமைப்பில் ஒரு மிதவைப் புள்ளி எண்ணை 10ன் அடுக்குகளில் கொண்டிருக்கும். மேலும், அதிகபட்சமாக துல்லியம் 64 இலக்கங்கள் வரை வரையறுக்கப்படலாம்.
real	இது float வகையைப் போன்றது. இருப்பினும், size செயலுருபை பயன்படுத்திக் கொள்ளாது. அதிகபட்சம், துல்லியம் 64 இலக்கங்கள் வரையறுக்கப்படலாம்.
double	இது real வகையைப் போன்றது. ஆனால் துல்லியம் 64 இலக்கங்களுக்கு கூடுதலாக இருக்கலாம்.

அட்டவணை *12.1*

12.7 SQL கட்டளைகளும் அதன் செயல்பாடுகளும்

தரவுகளை சேமித்து வைக்க ஒரே வழி அட்டவணை என்பதால், அனைத்து தகவல்களும் அட்டவணை வடிவில் ஒருங்கிணைக்கப்பட வேண்டும். SQL தரவுத்தளத்துடன் செயலாற்றுவதற்கான முன்வரையறுக்கப்பட்ட கட்டளைகளின் தொகுதியை வழங்குகிறது.

சிறப்புச் சொற்கள்	இவை SQL ல் சிறப்பு பொருளைக் கொண்ட கட்டளைகளாக அறியப்படுகிறது.
கட்டளைகள்	கூற்றுகள் என அறியப்படும் இவை பயனரால் தரவுத்தளத்திற்க கொடுக்கப்படும் அறிவுறுத்தல்களைக் குறிக்கும்.
Clauses	இவைச் சிறப்புச் சொல்லுடன் தொடங்கி, சிறப்புச் சொல்லையும், செயலுருபையும் கொண்டிருக்கும்.
செயலுருபுகள்	இவை clause ஐ நிறைவுப் பெற செய்ய கொடுக்கப்படும் மதிப்புகளை குறிக்கும்.

Chapter 12.indd 214 30-01-2020 20:45:16

12.7.1 DDL கட்டளைகள்

CREATE TABLE கட்டளைகள்

CREATE TABLE கட்டளையை பயன்படுத்தி ஒரு அட்டவணையை உருவாக்கலாம். ஒரு அட்டவணையை உருவாக்கும் போது, அதன் நெடுவரிசைகளுக்கு பெயரிட்டு, தரவு வகைகளையும், அளவினையும் குறிப்பிட வேண்டும். ஒவ்வொரு அட்டவணையும் குறைந்த பட்சம் ஒரு நெடுவரிசையை கொண்டிருக்க வேண்டும். CREATE TABLE கட்டளையின் தொடரியல்.

```
CREATE TABLE <table-name>
(<column name><data type>[<size>]
(<column name><data type>[<size>]......
);
```

மேற்கண்ட தொடரியலைப் பயன்படுத்தி, ஒரு வகுப்பில் பயிலும் மாணவர்களின் விவரங்களை ஒரு தரவுத்தளத்தில் சேமிக்கலாம். இதற்கு முதலில் நாம் ஒரு அட்டவணையை உருவாக்க வேண்டும். ஒரு மாணவர் குறித்த அட்டவணையில் மாணவர் தொடர்பான விவரங்களாக சேர்க்கை எண் (Admission number)- இதை சுருக்கி Admno என்றும், மாணவரின் பெயர் (Name), பாலினம் (gender), வயது (Age) போன்றவை எடுத்துக்கொள்ளலாம். Admno, Name, Gender, Age மற்றும் Place என்ற புலங்களை கொண்டுள்ள ஒரு அட்டவணையை நாம் உருவாக்கலாம்.

SQL கட்டளை பின்வருமாறு இருக்கும்:

```
CREATE TABLE Student
    (Admno    integer,
      Name  char(20),
      Gender  char(1),
      Age     integer,
      Place char(10),
      );
```

மேலே உள்ளது, எந்த கட்டுப்பாடுகளும் இல்லாத ஒரு எளிய அட்டவணை அமைப்பாகும். ஒரு அட்டவணையில் அமைந்திருக்கும் புலங்களுக்கான தரவுகளை சில கட்டுப்பாடுகளை ஏற்படுத்தி கட்டுப்படுத்தலாம். இது தரவுத்தளத்திலுள்ள தரவுகளின் துல்லியத்தன்மை மற்றும் நம்பகத்தன்மையை உறுதி செய்கிறது. கட்டுப்பாடுகள் நெடுவரிசை அல்லது அட்டவணை அளவிலானதாக இருக்கக்கூடும்.

குறிப்பு

கட்டுப்பாடுகள் என்பது ஒரு புலம் அல்லது புலங்களின் தொகுதிக்கு பயன்படுத்தக்கூடிய நிபந்தனையை குறிக்கும்.

நெடுவரிசை கட்டுப்பாடு	தனிப்பட்ட நெடுவரிசைக்கு மட்டுமே நெடுவரிசை கட்டுப்பாட்டை பயன்படுத்த முடியும்.
அட்டவணை கட்டுப்பாடு	அட்டவணை கட்டுப்பாடு ஒன்று அல்லது அதற்கு மேற்பட்ட நெடுவரிசைகளுக்கு பயன்படுத்த முடியும்.

Chapter 12.indd 215 30-01-2020 20:45:16

கட்டுப்பாட்டுடன் உருவாக்கப்பட்ட ஒரு அட்டவணையின் தொடரியல் கீழே கொடுக்கப்பட்டுள்ளது.

```
CREATE TABLE <table-name>
(<column name><data type>[<size>]<column constraint>,
(<column name><data type>[<size>]<column constraint>……
<table constraint>(<column name>,[<column name>….]).….
);
```

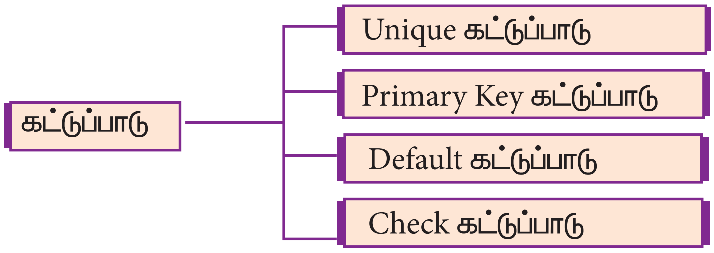
"NOT NULL" நெடுவரிசை கட்டுப்பாட்டுடன் கூடிய மாணவர் அட்டவணை கீழே தரப்பட்டுள்ளது. இந்த கட்டுப்பாடு ஒரு புலம் எப்பொழுதும் ஒரு மதிப்பைக் கொண்டிருக்க வேண்டும் என்பதை அறிவுறுத்துகிறது.

```
CREATE TABLE Student
(
Admno integer   NOT NULL PRIMARY KEY, → Primary Key constraint
Name char(20) NOT NULL,
Gender char(1),
Age integer,
Place char(10),
);
```

மேற்கண்ட கட்டளை அமைப்பை பயன்படுத்தி உருவாக்கப்பட்ட student என்ற அட்டவணையில் integer வகையை சார்ந்த Admno என்ற புலமும், char வகையை சார்ந்த Name என்ற புலமும், NOT NULL என்ற கட்டுப்பாட்டுடன் வரையறுக்கப்பட்டுள்ளது. இதன் பொருள் என்னவெனில், அந்த இரண்டு புலங்களும் கட்டாயமாக மதிப்புகளை கொண்டிருக்க வேண்டும். Gender, Age மற்றும் Place என்ற புலங்களுக்கு எந்த வித கட்டுப்பாடுகளும் இல்லை.

12.7.2 கட்டுப்பாடுகளின் வகைகள்

கட்டுப்பாடுகள் தரவுத்தள ஒருங்கிணைப்பை உறுதி செய்வதால், அவற்றை தரவுத்தள ஒருங்கிணைப்பு கட்டுப்பாடுகள் எனவும் அழைக்கலாம். கட்டுப்பாடுகள் பல வகைப்படும். அவை



12.7.2.1 Unique கட்டுப்பாடு

இந்த கட்டுப்பாடு குறிப்பிட்ட நெடுவரிசைகளில் எந்த இரு வரிசைகளும் ஒரே மதிப்பை கொண்டிருக்காது என்பதை உறுதி செய்கிறது. உதாரணத்திற்கு, student அட்டவணையில் Admno என்ற புலத்திற்கு UNIQUE கட்டுப்பாட்டை பயன்படுத்தி, ஒரே சேர்க்கை எண் இரு மாணவர்களுக்கு தரப்படாததை உறுதி செய்து கொள்ளலாம். இந்த கட்டுப்பாட்டை கீழ்க்கண்டவாறு பயன்படுத்திக் கொள்ளலாம்:

Chapter 12.indd 216 30-01-2020 20:45:16

```
CREATE TABLE Student
(
Admno integer NOT NULL UNIQUE, → Unique constraint
Name char (20) NOT NULL,
Gender char (1),
Age integer,
Place char (10),
);
```

UNIQUE கட்டுப்பாட்டை NOT NULL என்று அறிவிக்கப்பட்ட புலங்களுக்கு மட்டுமே பயன்படுத்த முடியும்.

ஒரு தனிப்பட்ட புலத்திற்கு இரு கட்டுப்பாடுகளை பயன்படுத்தினால் அப்புலம் பல கட்டுப்பாடுகளைக் கொண்ட புலம் என அழைக்கலாம். மேற்கண்ட கட்டணை அமைப்பில், NOT NULL மற்றும் UNIQUE எனும் பல கட்டுப்பாடுகள் எனும் பல கட்டுப்பாடுகள் Admno என்ற ஒரு தனித்த புலத்தில் பயன்படுத்தப்பட்டுள்ளது. கட்டுப்பாடுகள் இடைவெளிக் கொண்டு பிரிக்கப்பட்டு புல வரையறுப்பின் இறுதியில் காற்புள்ளி சேர்க்கப்படுகிறது. Admno என்ற புலத்திற்கு இந்த இரண்டு கட்டுப்பாடுகளை சேர்ப்பதால், இந்த புலம் ஏதேனும் சில மதிப்புகளைக் கொண்டிருக்க வேண்டும் அதே சமயம் இரட்டையான மதிப்புகள் இருத்தல் கூடாது.

12.7.2.2 Primary Key கட்டுப்பாடு

தரவுத்தளத்திலுள்ள ஒரு பதிவை தனித்தன்மையோடு அடையாளம் காட்ட ஒரு புலத்தினை primary key என்ற கட்டுப்பாட்டுடன் அறிவிக்க வேண்டும். இது unique கட்டுப்பாட்டினை போன்றே காணப்பட்டாலும் அட்டவணையின் ஒரு புலத்தை மட்டுமே primary key ஆக குறிப்பிட முடியும். primary key கட்டுப்பாடு வெற்று (NULL) மதிப்புகளை அனுமதிக்காது என்பதால், primary key என்ற கட்டுப்பாட்டுடன் அறிவிக்கப்படும் புலம், NOT NULL என்ற கட்டுப்பாட்டையும் கொண்டிருக்க வேண்டும்.

மாணவர் அட்டவணையில் primary key கட்டுப்பாட்டை பயன்படுத்துதலை விளக்கும் எடுத்துக்காட்டு:

```
CREATE TABLE Student
(
Admno integer    NOT NULL PRIMARY KEY, → Primary Key constraint
Name char(20) NOT NULL,
Gender char(1),
Age integer,
Place char(10),
);
```

மேற்கண்ட எடுத்துக்காட்டில், Admno என்ற புலம் primary key என்று வரையறுக்கப்பட்டுள்ளதால், ஒரு பதிவை தனித்தன்மையோடு அடையாளம் காட்ட முடிகிறது. மேலும் NOT NULL என்றும் வரையறுக்கப்பட்டுள்ளதால், இந்த புலத்தின் மதிப்பு காலியாக இருக்கக் கூடாது.

12.7.2.3 DEFAULT கட்டுப்பாடு

DEFAULT கட்டுப்பாடு ஒரு புலத்தில் முன்னியல்பு மதிப்புகளை இருத்தி வைக்க பயன்படுகிறது. DEFAULT கட்டுப்பாட்டை கொண்ட குறிப்பிட்ட புலத்திற்கு மதிப்பு கொடுக்காவிடில், தானாகவே முன்னியல்பு

Chapter 12.indd 217 30-01-2020 20:45:16

மதிப்பு அந்த புலத்தில் இருத்தப்பட்டு விடும். மாணவர் அட்டவணையில் Default கட்டுப்பாடு பயன்படுத்துதலை விளக்கும் எடுத்துக்காட்டு:

```
CREATE TABLE Student
(
Admno integer    NOT NULL  PRIMARY KEY,
Name char(20) NOT NULL,
Gender char(1),
Age integer  DEFAULT "17", → Default Constraint
Place char(10),
);
```

மேற்கண்ட எடுத்துக்காட்டில், "Age" என்ற புலத்திற்கு 17 என்று முன்னியல்பு மதிப்பு கொடுக்கப்பட்டுள்ளதால், அந்த புலத்தில் பயனரால் எந்த மதிப்பும் உள்ளிடப்படவில்லை எனில், "Age" என்ற புலம் தானாகவே 17 என்ற மதிப்பை எடுத்துக் கொள்ளும்.

12.7.2.4 Check கட்டுப்பாடு

இந்தக் கட்டுப்பாடு ஒரு புலத்திற்கான மதிப்பின் வரம்பை நிர்ணயிக்க உதவுகிறது. CHECK கட்டுப்பாட்டை ஒரு நெடுவரிசைக்கு வரையறுத்தால், அப்புலத்தில் வரையறுக்கப்பட்ட மதிப்புகளை மட்டுமே உள்ளிட அனுமதிக்கும்.

மாணவர் அட்டவணையில் CHECK கட்டுப்பாட்டை பயன்படுத்துதலை விளக்கும் எடுத்துக்காட்டு

```
CREATE TABLE Student
(
Admno integer    NOT NULL PRIMARY KEY
Name char(20) NOT NULL,
Gender char(1),
Age integer (CHECK Age <=19), → Check Constraint
Place char(10),
);
```

மேற்கண்ட எடுத்துக்காட்டில், Age என்ற புலத்தில் CHECK கட்டுப்பாட்டை பயன்படுத்தி, Age என்ற புலத்திற்கு உள்ளிடும் மதிப்பு 19க்கு குறைவாகவோ அல்லது சம்மாகவோ இருக்க வேண்டும் என வரையறுக்கப்படுகிறது.

குறிப்பு

CHECK கட்டுப்பாடு ஒப்பீட்டு மற்றும் தருக்க செயற்குறிகளை நிபந்தனையை வரையறுக்க பயன்படுத்திக் கொள்ளலாம்.

12.7.2.5 அட்டவணை கட்டுப்பாடு

ஒரு அட்டவணையில் உள்ள புலங்களின் குழுவிற்கு கட்டுப்பாட்டைப் பயன்படுத்தினால், அதனை அட்டவணைக் கட்டுப்பாடு என்கிறோம். பொதுவாக, அட்டவணை கட்டுப்பாடு அட்டவணை வரையறுப்பின் இறுதியில் குறிப்பிடப்படும். Admno, Firstname, Lastname, Gender, Age, Place என்ற புலங்களை கொண்ட student 1 என்ற ஒரு புதிய அட்டவணையை எடுத்துக் கொள்வோம்.

Chapter 12.indd 218 30-01-2020 20:45:16

```
CREATE TABLE Student1
(
Admno integer NOT NULL,
Firstname char(20),
Lastname char(20),
Gender char(1),
Age integer,
Place char(10),
PRIMARY KEY (Firstname, Lastname) → Table constraint
);
```

மேற்கண்ட எடுத்துக்காட்டில், primary key என்று வரையறுக்கப்பட்டுள்ள Firstname, Lastname என்ற இருபுலங்கள், அட்டவணை கட்டுப்பாடுடையதாக அமைந்துள்ளன.

12.7.3 DML கட்டளைகள்

ஒரு அட்டவணையின் திட்டமைப்பு (அல்லது) வடிவமைப்பு உருவாக்கப்பட்டவுடன், அதில் தரவுகளை சேர்க்கலாம். அட்டவணையில் வரிசைகளை சேர்த்தல், நீக்குதல், புதுப்பித்தல் போன்ற செயற்பாடுகளுக்கு DML கட்டளைகள் பயன்படுகிறது.

12.7.3.1 INSERT கட்டளை

INSERT கட்டளை தரவுத்தளத்தில் புதிய தரவை சேர்ப்பதற்கும், அல்லது ஒரு அட்டவணையில் புதிய பதிவுகளை சேர்ப்பதற்கு உதவுகிறது. இந்த கட்டளையை எவ்வாறு பயன்படுத்துவது என்பது கீழே குறிப்பிடப்பட்டுள்ளது:

INSERT INTO <table-name> [column-list] VALUES (values);

INSERT INTO Student (Admno, Name, Gender, Age, Place)

VALUES (100, 'Ashish', 'M', 17, 'Chennai');

INSERT INTO Student (Admno, Name, Gender, Age, Place)

VALUES (101, 'Adarsh', 'M', 18, 'Delhi');

அட்டவணையில் இரு புதிய பதிவுகள் சேர்க்கப்பட்டுள்ளதை கீழ்காணலாம்:

Admno	Name	Gender	Age	Place
100	Ashish	M	17	Chennai
101	Adarsh	M	18	Delhi

மதிப்புகளின் வரிசைமுறை CREATE TABLE கட்டளையில் உள்ள நெடுவரிசையின் வரிசை முறையுடன் பொருந்தி இருக்க வேண்டும். அனைத்து நெடுவரிசைகளுக்கும் தரவு சேர்க்கபட வேண்டுமாயின், நெடுவரிசை பெயர்களைக் குறிப்பிடுவது கட்டாயமில்லை. மாணவர் அட்டவணையில் தரவுகளை பின்வரும் கட்டளையின் மூலமும் உள்ளிடலாம்.

INSERT INTO Student VALUES (102, 'Akshith', 'M', 17, 'Bangalore');

102	Akshith	M	17	Bangalore

மேற்கண்ட கட்டளை மாணவர் அட்டவணையில் ஒரு பதிவை சேர்க்கிறது.

ஒரு பதிவினுடைய சில நெடுவரிசைகளுக்கு மட்டும் தரவை சேர்க்க, நெடுவரிசையின் பெயரையும், அதற்கான தரவையும் கீழ்க்கண்டவாறு தரலாம்.

Chapter 12.indd 219 30-01-2020 20:45:16

INSERT INTO Student(Admno, Name, Place) VALUES (103, 'Ayush', 'Delhi');

103	Ayush	M	18	Delhi

மேற்கண்ட கட்டளை பின்வரும் பதிவினை Gender என்ற புலத்திற்கு முன்னியல்பு மதிப்பான 'M' என்றும், Age என்ற புலத்திற்கு முன்னியல்பு மதிப்பான 18 என்றும் சேர்க்கும்.

INSERT INTO Student (Admno, Name, Place) VALUES (104, 'Abinandh', 'Chennai');

104	Abinandh	M	18	Chennai

மாணவர் அட்டவணை பின்வரும் தரவைக் கொண்டிருக்கும்:

Admno	Name	Gender	Age	Place
100	Ashish	M	17	Chennai
101	Adarsh	M	18	Delhi
102	Akshith	M	17	Bangalore
103	Ayush	M	18	Delhi
104	Abinandh	M	18	Chennai

INSERT கட்டளையில் கொடுக்கப்படாத புலங்கள், முன்னியல்பு மதிப்புகள் அதற்கு வரையறுக்கப்பட்டிருந்தால் அவற்றை எடுத்துக் கொள்ளும், இல்லையேல் NULL மதிப்பை இருத்திக் கொள்ளும்.

உங்களுக்குத் தெரியுமா?

INSERT கட்டளையில், விடுப்பட்ட புலங்கள் வரையறுக்கப்பட்ட முன்னியல்பு மதிப்பையோ அல்லது NULL மதிப்பையோ கொண்டிருக்கும்.

12.7.3.2 DELETE கட்டளை

DELETE கட்டளை அட்டவணையிலிருந்து ஒன்று அல்லது அதற்கு மேற்பட்ட பதிவுகளை நீக்குகிறது. இது தனிப்பட்ட புலத்தை மட்டும் நீக்காமல், வரிசை முழுவதையும் நீக்குவதால், புலத்திற்கான செயலுருபு தேவையில்லை.

DELETE கட்டளையின் பயன்பாடு பின்வருமாறு :

DELETE FROM table-name WHERE condition;

உதாரணத்திற்கு, **104** என்ற சேர்க்கை எண் கொண்ட பதிவை நீக்குவதற்கான கட்டளை பின்வருமாறு:

DELETE FROM Student WHERE Admno=104;

104	Abinandh	M	18	Chennai

மேற்கண்ட பதிவு மாணவர் அட்டவணையிலிருந்து நீக்கப்படும்.

அட்டவணையில் உள்ள அனைத்து வரிசைகளையும் நீக்க, பயன்படும் கட்டளை :

DELETE * FROM Student;

Chapter 12.indd 220 30-01-2020 20:45:16

இப்பொழுது அட்டவணை வெறுமையாக காணப்படும். DROP கட்டளையை (12.7.4.3 பகுதியில் பார்க்கலாம்) பயன்படுத்தி. இந்த அட்டவணையை முழுவதுமாக அழிக்க முடியும்.

12.7.3.3 UPDATE கட்டளை

ஒரு தரவுத்தளத்தில் உள்ள சில அல்லது அனைத்து தரவு மதிப்புகளையும் UPDATE கட்டளை புதுப்பித்தல் செய்யும்(புதுப்பிக்கும்). அது அட்டவணையில் உள்ள ஒன்று அல்லது அதற்கு மேற்பட்ட பதிவுகளையும் புதுப்பிக்க முடியும். UPDATE கட்டளை, WHERE clause ஐ பயன்படுத்தி மாற்றப்பட வேண்டிய வரிசைகளையும், SET சிறப்புச் சொல்லைப் பயன்படுத்தி புதிய தரவையும் குறிப்பிடுகிறது. UPDATE கட்டளையை பின்வருமாறு பயன்படுத்தலாம்:

> ***UPDATE <table-name> SET column-name = value, column-name = value,… WHERE condition;***

உதாரணத்திற்கு, பின்வரும் புலங்களை புதுப்பிக்க:

UPDATE Student SET Age = 20 WHERE Place = "Bangalore";

மேற்கண்ட கட்டளை **"Bangalore"** என்ற இடத்தில் இருக்கும் மாணவர்களுடைய வயதை **20** என மாற்றி அமைக்கும். புதுப்பிக்கப்பட்ட அட்டவணை பின்வருமாறு:

Admno	Name	Gender	Age	Place
100	Ashish	M	17	Chennai
101	Adarsh	M	18	Delhi
102	Akshith	M	20	Bangalore
103	Ayush	M	18	Delhi

ஒன்றுக்கு மேற்பட்ட புலங்களை புதுப்பிக்க, SET clause பயன்படுத்தி அவற்றிற்கான மதிப்புகளை காற்புள்ளியால் பிரித்து குறிப்பிடலாம். உதாரணத்திற்கு, ஒரு மாணவர் அட்டவணையில், பல்வேறு புலங்களை புதுப்பித்தலுக்கான கட்டளை பின்வருமாறு :

UPDATE Student SET Age=18, Place = 'Chennai' WHERE Admno = 102;

102	Akshith	M	18	Chennai

மேற்கண்ட கட்டளை கீழ்க்கண்ட முறையில் பதிவை மாற்றி அமைக்கிறது.

Admno	Name	Gender	Age	Place
100	Ashish	M	17	Chennai
101	Adarsh	M	18	Delhi
102	Akshith	M	18	Chennai
103	Ayush	M	18	Delhi

12.7.4 சில கூடுதல் DDL கட்டளைகள்

12.7.4.1 ALTER கட்டளை

ALTER கட்டளையைப் பயன்படுத்தி அட்டவணை வடிவமைப்பில் நெடுவரிசையை சேர்த்தல், ஏற்கனவே இருக்கும் புலத்திற்கு மறு பெயரிடல், எந்த ஒரு நெடுவரிசையின் தரவு வகையையோ அல்லது அளவினையோ மாற்றி அமைத்தல் நெடுவரிசையை அட்டவணையிலிருந்து நீக்குதல் போன்ற மாற்றங்களை செய்யலாம். அது கீழ்க்காணும் முறையில் பயன்படுத்தப்படுகிறது:

ALTER TABLE <table-name> ADD <column-name><data type><size>;

மாணவர் அட்டவணையில் 'char' என்ற தரவுவகை சார்ந்த "address" என்ற நெடுவரிசையை சேர்க்க, கட்டளையமைப்பு

ALTER TABLE Student ADD Address char;

அட்டவணையில் ஏற்கனவே இருக்கும் நெடுவரிசையை மாற்றியமைக்க, ALTER TABLE கட்டளை MODIFY clause உடன் பயன்படுத்தப்படுகிறது.

ALTER <table-name> MODIFY<column-name><data type><size>;

ALTER TABLE Student MODIFY Address char (25);

மேற்கண்ட கட்டளை மாணவர் அட்டவணையில் உள்ள address நெடுவரிசை இப்பொழுது 25 எழுத்துக்கள் வரை தேக்கி வைத்துக் கொள்ளும் வகையில் மாற்றி அமைக்கும்.

ALTER கட்டளை கீழ்க்காணும் முறையில் ஏற்கனவே உள்ள நெடுவரிசைக்கு மறுபெயரிட பயன்படுகிறது:

ALTER <table-name> RENAME old-column-name TO new-column-name;

உதாரணத்திற்கு, address என்ற பெயரைக் கொண்ட நெடுவரிசைக்கு city என்று மறுபெயரிட உதவும் கட்டளை :

ALTER TABLE Student RENAME Address TO City;

ஒரு நெடுவரிசை அல்லது அனைத்து நெடுவரிசைகளையும் நீக்க, ALTER கட்டளையைப் பயன்படுத்தலாம். உதாரணத்திற்கு, ALTER TABLE-ன் DROP COLUMN ஐ பயன்படுத்தி, குறிப்பிட்ட நெடுவரிசையை நீக்கலாம். கட்டளை:

ALTER <table-name> DROP COLUMN <column-name>;

மாணவர் அட்டவணையில் city என்ற நெடுவரிசையை நீக்க, பயன்படும் கட்டளை:

ALTER TABLE Student DROP COLUMN City;

12.7.4.2 TRUNCATE கட்டளை

TRUNCATE கட்டளை அட்டவணையிலுள்ள அனைத்து வரிசைகளையும் நீக்கப் பயன்படுகிறது. அட்டவணையின் வடிவமைப்பு மாறாமல், ஒதுக்கீடு செய்யப்பட்ட இடம் விடுவிக்கப்படும் TRUNCATE கட்டளையின் தொடரியல்

TRUNCATE TABLE table-name;

உதாரணத்திற்கு, மாணவர் அட்டவணையில் உள்ள அனைத்து பதிவுகளையும், அட்டவணையையும் நீக்குவதற்கான கட்டளை :

TRUNCATE TABLE Student;

மாணவர் அட்டவணை நீக்கப்பட்டு, நினைவகப்பகுதி விடுவிக்கப்படுகிறது.

12.7.4.3 DROP TABLE கட்டளை

DROP TABLE கட்டளை தரவுத்தளத்திலிருந்து ஒரு அட்டவணையை நீக்கப் பயன்படுகிறது. ஒரு அட்டவணையை **DROP** செய்தால், அட்டவணையில் உள்ள அனைத்து வரிசைகளும் நீக்கப்பட்டு, அட்டவணையின் வடிவமைப்பு தரவுத்தளத்திலிருந்து நீக்கப்படும். அட்டவணையை **DROP** செய்த பின்பு, அதனை திரும்ப பெற இயலாது. ஆதலால், **DROP TABLE** கட்டளையை பயன்படுத்தும் போது எச்சரிக்கையாக இருத்தல் வேண்டும். ஒரு அட்டவணையை **DROP** செய்ய வேண்டுமெனில், அது காலி அட்டவணையாக இருக்க வேண்டும். அட்டவணையின் அனைத்து வரிசைகளையும் **DELETE** கட்டளை கொண்டு நீக்கப்பட்டிருக்க வேண்டும்.

அனைத்து வரிசைகளையும் நீக்க, தேவையான கட்டளை:

DELETE * FROM Student;

அனைத்து வரிசைகளும் நீக்கப்பட்டப் பின்பு, கீழ்க்காணும் முறையில் DROP TABLE கட்டளையை பயன்படுத்தி அட்டவணையை நீக்கலாம்.

DROP TABLE table-name;

உதாரணத்திற்கு மாணவர் அட்டவணையை நீக்க:

DROP TABLE Student;

DELETE, TRUNCATE AND DROP கூற்று:

DELETE	**DELETE** கட்டளை **WHERE** clause-ல் கொடுக்கப்பட்டிருக்கும் நிபந்தனையின் அடிப்படையில், அட்டவணையின் குறிப்பிட்ட வரிசைகளை மட்டும் நீக்கும் அல்லது எந்த வித நிபந்தனையும் கொடுக்கப்படவில்லையெனில், அட்டவணையின் அனைத்து வரிசைகளையும் நீக்கி விடும். ஆனால் அட்டவணைக்கென ஒதுக்கப்பட்ட இடத்தை விடுவிக்காது.
TRUNCATE	TRUNCATE கட்டளை அட்டவணையின் அனைத்து வரிசைகளையும் நீக்கி, வடிவமைப்பை மாற்றாமல், அட்டவணைக்கென ஒதுக்கீடு செய்யப்பட்ட இடத்தை விடுவிக்கும்.
DROP	DROP கட்டளை தரவுத்தளத்திலிருந்து ஒரு உறுப்பை நீக்க பயன்படுகிறது. ஒரு அட்டவணையை DROP செய்தால், அட்டவணையில் உள்ள அனைத்து வரிசைகளும் நீக்கப்பட்டு, அட்டவணையின் வடிவமைப்பு தரவுத்தளத்திலிருந்து நீக்கப்படும். ஒரு அட்டவணையை DROP செய்துவிட்ட பின்பு, அதனை திரும்பப் பெற இயலாது.

12.7.5 DQL கட்டளை – SELECT கட்டளை

SQL-ல் செயலாற்றும் போது, வினவல்களை உருவாக்குவதும், தரவை மீட்டெடுப்பதும் மிக முக்கியமான பணிகளில் ஒன்றாகும். ஒரு வினவல் என்பது தரவுத்தள அட்டவணையிலிருந்து நாம் விரும்பும் தீர்வை பெற கொடுக்கப்படும் கட்டளையாகும். SQL SELECT கட்டளை, தரவுத்தள அட்டவணையில் உள்ள

Chapter 12.indd 223 30-01-2020 20:45:17

தரவினை வினவலாகவோ அல்லது மீட்டெடுக்கவோ பயன்படும் கட்டளையாகும். இது ஒன்று அல்லது அதற்கு மேற்பட்ட அட்டவணைகளிலிருந்து பதிவுகளின் துணைக் கூறுகளை மீட்டெடுக்கப் பயன்படுகிறது. SELECT கட்டளையை பல வடிவங்களில் பயன்படுத்தலாம்.

SELECT கட்டளையின் தொடரியல் :

SELECT <column-list>FROM<table-name>;

- table-name என்பது தகவல் பெறப்பட்ட அட்டவணையின் பெயர்.
- Column-list என்பது தரவு பெறப்பட்ட ஒன்று அல்லது அதற்கு மேற்பட்ட நெடுவரிசைகளை உள்ளடக்கியது.

உதாரணமாக மாணவர் அட்டவணையில் இருந்து சேர்க்கை எண் மற்றும் மாணவர்களின் பெயரை மட்டுமே பார்க்க, கட்டளை பின்வருமாறு கொடுக்கப்பட்டுள்ளது:

மாணவர் அட்டவணை பின்வரும் தரவைக் கொண்டிருந்தால்:

Admno	Name	Gender	Age	Place
100	Ashish	M	17	Chennai
101	Adarsh	M	18	Delhi
102	Akshith	M	17	Bangalore
103	Ayush	M	18	Delhi
104	Abinandh	M	18	Chennai
105	Revathi	F	19	Chennai
106	Devika	F	19	Bangalore
107	Hema	F	17	Chennai

SELECT Admno, Name FROM Student;

மேற்கண்ட **SELECT** கட்டளை பின்வரும் தகவலை வெளிக்காட்டும்:

Admno	Name
100	Ashish
101	Adarsh
102	Akshith
103	Ayush
104	Abinandh
105	Revathi
106	Devika
107	Hema

அட்டவணையின் அனைத்து புலங்களையும், வரிசைகளையும் பார்க்க, **SELECT** கட்டளை இவ்வாறு அமையும்.

SELECT * FROM STUDENT;

Chapter 12.indd 224 30-01-2020 20:45:17

12.7.5.1 சிறப்புச் சொல் DISTINCT

DISTINCT சிறப்புச் சொல்லுடன் SELECT கட்டளையைப் பயன்படுத்தி ஒரு அட்டவணையில் இரட்டிப்பு வரிசைகளை உருவாகாமல் தடுக்கலாம். இது தரவு இரட்டித்தலைத் தவிர்க்கின்றது. உதாரணத்திற்கு,

SELECT * DISTINCT place FROM student; என்ற கட்டளை பின்வரும் தரவை வெளிக்காட்டும்:

SELECT * FROM Student;

Place
Chennai
Bangalore
Delhi

மேற்கண்ட வெளியீட்டில், place என்னும் புலத்தில் இரட்டிப்பு வரிசைகள் இல்லாத்தைக் காணலாம். DISTINCT என்ற சிறப்புச் சொல்லை பயன்படுத்தும் போது NULL மதிப்புகள் அதிக எண்ணிக்கையில் இருப்பினும், ஒரே ஒரு NULL மதிப்பு மட்டுமே திருப்பி அனுப்பப்படும்.

12.7.5.2 சிறப்புச் சொல் ALL

ALL என்ற சிறப்புச் சொல் இரட்டிப்பு வரிசைகளையும் இருத்தி வைக்கிறது. அது இரட்டிப்பு பதிவுகளை கருத்தில் கொள்ளாது, அட்டவணையின் ஒவ்வொரு வரிசையையும் வெளிக்காட்டும்.

SELECT ALL Place FROM Student;

மேற்கண்ட கட்டளை அட்டவணையில் உள்ள ஒவ்வொரு வரிசையிலும் உள்ள place என்ற புலத்தின் அனைத்து மதிப்புகளையும்(இரட்டிப்புப் பதிவுகளையும் உள்ளடக்கி) வெளிக்காட்டும்.

Place
Chennai
Delhi
Bangalore
Delhi
Chennai
Chennai
Bangalore
Chennai

SELECT கட்டளையிலுள்ள **WHERE** clause, விரும்பிய தீர்வை பெறுவதற்கான விதிமுறைகளை குறிப்பிடுகிறது. WHERE clause யுடனான SELECT கட்டளையின் பொது வடிவம்:

SELECT <column-name>[,<column-name>,….] FROM <table-name>WHERE condition>;

உதாரணத்திற்கு, மாணவர்களின் சேர்க்கை எண்ணையும், சென்னையை சேர்ந்த மாணவர்களின் பெயர்களை மட்டும் வெளிக்காட்ட, SELECT கட்டளை கீழ்க்காணும் முறையில் பயன்படுத்தப்படுகிறது:

SELECT Admno, Name, Place FROM Student WHERE Place =”Chennai”;

Admno	Name	Place
100	Ashish	Chennai
104	Abinandh	Chennai
105	Revathi	Chennai
107	Hema	Chennai

SELECT Admno, Name, Age FROM Student WHERE Age >= 18;

Admno	Name	Age
101	Adarsh	18
103	Ayush	18
104	Abinandh	18
105	Revathi	19
106	Devika	19

ஒப்பீட்டு இயக்கிகள் =, <, <=, >, >=, <> போன்றவை இரண்டு மதிப்புகளை ஒப்பீட SELECT கட்டளையுடன் WHERE clause பயன்படுத்தலாம். WHERE clauseல் தேடல் நிபந்தனைகளை OR, AND மற்றும் NOT போன்ற தருக்க செயற்குறிகள் ஒன்றிணைக்க பயன்படுகின்றன. உதாரணத்திற்கு,

SELECT Admno, Name, Age, Place FROM Student WHERE (Age>=18 AND Place = "Delhi");

Admno	Name	Age	Place
101	Adarsh	18	Delhi
103	Ayush	18	Delhi

The **SELECT** கட்டளையை கீழ்க்காணும் முறைகளிலும் பயன்படுத்தலாம்:

SELECT Admno, Name, Age, Place FROM Student WHERE (Age>=18 OR Place ="Delhi");

SELECT Admno, Name, Place FROM Student WHERE (NOT Place ="Delhi");

12.7.5.3 BETWEEN மற்றும் NOT சிறப்புச் சொற்கள்

BETWEEN என்ற சிறப்புச் சொல் மதிப்புகளின் பரப்பை வரையறுக்கின்றது, இந்த பரப்பினுள் உள்ள மதிப்புகள் கொண்ட வரிசைகளை உள்ளடக்கும். மதிப்புகளின் பரப்பான அதிகபட்ச மதிப்பு மற்றும் குறைந்தபட்ச மதிப்பிற்கு இடையே உள்ள நிபந்தனையை உள்ளடக்கும்.

SELECT Admno, Name, Gender, Age, Place FROM Student WHERE Age BETWEEN 18 AND 19;

Admno	Name	Gender	Age	Place
101	Adarsh	M	18	Delhi
103	Ayush	M	18	Delhi
104	Abinandh	M	18	Chennai
105	Revathi	F	19	Chennai
106	Devika	F	19	Bangalore

Chapter 12.indd 226 30-01-2020 20:45:17

BETWEEN இயக்கியின் எதிர் இயக்கியாக NOT BETWEEN செயல்படும், நிபந்தனையை நிறைவு செய்யாத பதிவுகளை இது வெளிக்காட்டும்.

SELECT Admno, Name, Age FROM Student WHERE Age NOT BETWEEN 18 AND 19;

Admno	Name	Age
100	Ashish	17
102	Akshith	17
107	Hema	17

12.7.5.4 IN சிறப்புச் சொல்

IN சிறப்புச் சொல் பதிவின் மதிப்புகளுடன் பொருந்தக்கூடிய வகையில் மதிப்புகளின் பட்டியலைக் குறிப்பிடப் பயன்படுகிறது. வேறு சொற்களில் கூறுவதெனில், இது ஒன்றுக்கு மேற்பட்ட மதிப்புகளுடன் ஒரு நெடுவரிசையை ஒப்பிடப் பயன்படுகிறது. இது OR நிபந்தனைக்கு நிகரானது.

உதாரணத்திற்கு :

SELECT Admno, Name, Place FROM Student WHERE Place IN ("Chennai", "Delhi");

Admno	Name	Place
100	Ashish	Chennai
101	Adarsh	Delhi
103	Ayush	Delhi
104	Abinandh	Chennai
105	Revathi	Chennai
107	Hema	Chennai

NOT IN சிறப்புச் சொல் பட்டியலுடன் பொருந்தாத மதிப்புகளைக் கொண்ட பதிவுகளை மட்டுமே வெளிக்காட்டும். உதாரணத்திற்கு,

SELECT Admno, Name, Place FROM Student WHERE Place NOT IN ("Chennai", "Delhi");

என்ற கட்டளை "Chennai" மற்றும் "Delhi" ஆகிய இடங்களை தவிர மற்ற மாணவர்களின் பட்டியலை வெளியிடும்.

Admno	Name	Place
102	Akshith	Bangalore
106	Devika	Bangalore

NULL மதிப்பு :

ஒரு அட்டவணையில் **NULL** மதிப்பையுடைய புலத்தை WHERE clauseல் **IS NULL** என்பதைப் பயன்படுத்தி தேடலாம். உதாரணத்திற்கு, age என்ற புலத்தில் மதிப்புகள் கொடுக்கப்படாத அனைத்து மாணவர்களின் பட்டியலை வெளியிட, பயன்படுத்த வேண்டிய கட்டளை:

SELECT * FROM Student WHERE Age IS NULL;

குறிப்பு

IS NOT NULL என்பதை பயன்படுத்தி அட்டவணையில் உள்ள NULL அல்லாத மதிப்புகளை பட்டியலிடலாம்.

12.7.5.5 ORDER BY clause

SQL-ன் ORDER BY clause பயன்படுத்தி தரவுகளை ஒன்று அல்லது அதற்கு மேற்பட்ட நெடுவரிசைகளின் அடிப்படையில் ஏறுவரிசையிலோ அல்லது இறங்கு வரிசையிலோ வரிசைப்படுத்தலாம்.

1. கொடாநிலையாக, **ORDER BY** தரவை ஏறுவரிசையில் வரிசைப்படுத்தும்.

2. தரவினை **DESC** என்ற சிறப்புச் சொல்லைப் பயன்படுத்தி இறங்கு வரிசையிலும்,

ASC என்ற சிறப்புச் சொல்லைப் பயன்படுத்தி ஏறுவரிசையிலும் வரிசைப்படுத்தலாம்.

ORDER BY ஐ இவ்வாறு பயன்படுத்தலாம் :

```
SELECT <column-name>[,<column-name>,….] FROM <table-name>ORDER BY
<column1>,<column2>,…ASC| DESC ;
```

உதாரணத்திற்கு :

மாணவர்களின் பெயரை அகர வரிசைப்படுத்திய பட்டியலை வெளியிட பயன்படும் கட்டளையமைப்பு

SELECT * FROM Student ORDER BY Name;

வரிசைப்படுத்தப்பட்ட மாணவர் அட்டவணை பின்வருமாறு:

Admno	Name	Gender	Age	Place
104	Abinandh	M	18	Chennai
101	Adarsh	M	18	Delhi
102	Akshith	M	17	Bangalore
100	Ashish	M	17	Chennai
103	Ayush	M	18	Delhi
106	Devika	F	19	Bangalore
107	Hema	F	17	Chennai
105	Revathi	F	19	Chennai

குறிப்பு

ORDER BY clause பயன்படுத்தினால் அட்டவணைக்கு எந்த பாதிப்பையும் ஏற்படுத்தாது.

12.7.5.6 WHERE clause

WHERE clause பதிவுகளை வடிகட்டுதலுக்கு பயன்படுகிறது. இது கொடுக்கப்பட்ட நிபந்தனையை நிறைவு செய்யும் பதிவுகளை மட்டும் பிரித்தெடுக்க உதவுகிறது. உதாரணத்திற்கு, மாணவர் அட்டவணையில்

18 மற்றும் அதற்கு மேற்பட்ட வயதுடைய மாணவர்களின் பட்டியலை, அகர வரிசைப்படுத்தி ஏறுவரிசையில் வெளியிடும் கட்டளை பின்வருமாறு:

SELECT * FROM Student WHERE Age>=18 ORDER BY Name;

Admno	Name	Gender	Age	Place
104	Abinandh	M	18	Chennai
101	Adarsh	M	18	Delhi
103	Ayush	M	18	Delhi
106	Devika	F	19	Bangalore
105	Revathi	F	19	Chennai

18 மற்றும் அதற்கு மேற்பட்ட வயதுடைய மாணவர்களின் பட்டியலை, அகர வரிசைப்படி இறங்கு வரிசையில், வெளியிடும் கட்டளை பின்வருமாறு :

SELECT * FROM Student WHERE Age>=18 ORDER BY Name DESC;

Admno	Name	Gender	Age	Place
105	Revathi	F	19	Chennai
106	Devika	F	19	Bangalore
103	Ayush	M	18	Delhi
101	Adarsh	M	18	Delhi
104	Abinandh	M	18	Chennai

குறிப்பு

பற்பல புலங்களின் மீதும் வரிசையாக்கம் செய்ய்யலாம்.

12.7.5.7 GROUP BY clause

SELECT கூற்றுடன் **GROUP BY** clause பயன்படுத்தி வரிசைகள் அல்லது நெடுவரிசைகளில் ஒரே மாதிரியான மதிப்புகளில் கொண்ட மாணவர்களை ஒரு தொகுதியாக பிரிக்கலாம் அல்லது ஒரு அட்டவணையை குழுக்களாக பிரிக்கலாம். உதாரணத்திற்கு ஒரு வகுப்பில உள்ள ஆண் மாணவர்கள் அல்லது பெண் மாணவிகளின் எண்ணிக்கையைத் தெரிந்துக் கொள்ள, GROUP BY ஐ பயன்படுத்தலாம்.

GROUP BY clause ன் தொடரியல்

SELECT <column-names> FROM <table-name> GROUP BY <column-name>HAVING [condition];

மேற்கண்ட கட்டளையை மாணவர் அட்டவணையில் பயன்படுத்த :

SELECT Gender FROM Student GROUP BY Gender;

மேற்கூறிய கட்டளையின் வெளியீடு பின்வருமாறு:

Gender
M
F

இரண்டு முடிவுகள் மட்டுமே திரும்ப பெறப்பட்டுள்ளது என்பதை கருத்தில் கொள்ளவும். இதன் காரணம் என்னவெனில், நாம் 'ஆண்' மற்றும் 'பெண்' என்ற இரு பாலினங்களையே பெற்றுள்ளோம். GROUP BY clause அனைத்து 'M' மாணவர்களையும் தொகுத்து, ஒரு வரிசையை மட்டும் வெளியிட்டுள்ளது. இது போன்றே 'F' மாணவர்களுக்கும் வெளியீடு அமையும்.

உதாரணத்திற்கு, மாணவர் அட்டவணையில் ஆண், பெண் மாணவர்களின் எண்ணிக்கையை கண்டறிய கீழ்காணும் கட்டளை தரப்பட்டுள்ளது :

SELECT Gender, count(*) FROM Student GROUP BY Gender;

Gender	count(*)
M	5
F	3

குறிப்பு

NULL மதிப்புகளையும் சேர்த்துக்கொள்ள COUNT உடன் * என்ற குறியீடு பயன்படுத்தப்பட்டுள்ளது.

மேற்கண்ட SELECT கூற்ற ஆண் மற்றும் பெண் மாணாக்கர்களின் எண்ணிக்கையை வெளியீடும்.

12.7.5.8 HAVING clause

SELECT கட்டளையுடன் HAVING மற்றும் GROUP BY Clause-க்களை பயன்படுத்தி குழுக்களுக்கு நிபந்தனையை இருத்தலாம் மற்றும் பயனர் ஒன்றிணைந்த செயற்கூறுகளையும் பயன்படுத்தலாம்.

உதாரணத்திற்கு, Chennai ஐ சேர்ந்த ஆண் மற்றும் பெண் மாணாக்கர்களின் எண்ணிக்கையை கண்டறிய

SELECT Gender , count(*) FROM Student GROUP BY Gender HAVING Place = 'Chennai';

Gender	count(*)
M	2
F	2

மேற்கண்ட வெளியீடு மாணவர் அட்டவணையில் சென்னையை சேர்ந்த ஆண் மற்றும் பெண் மாணாக்கர்களின் எண்ணிக்கையை காண்பிக்கிறது.

Chapter 12.indd 230 30-01-2020 20:45:17

12.7.6 TCL commands

12.7.6.1 COMMIT command

COMMIT கட்டளை, தரவுத்தளத்திலுள்ள எந்த ஒரு நடவடிக்கையும் நிரந்தரமாக சேமிக்கப்பயன்படுகிறது. **INSERT, UPDATE, DELETE** போன்ற DML கட்டளைகளைப் பயன்படுத்தி செய்யப்பட்ட மாற்றங்கள் நிரந்தரமல்ல. SQL தூண்டுகுறியில் COMMIT கட்டளையை கொடுத்த பிறகுதான், இது நிரந்தரமாக குறிக்கப்படுகிறது. COMMIT கட்டளையை கொடுத்து விட்டால், செய்யப்பட்ட மாற்றங்கள் திரும்பப் பெற இயலாது. **COMMIT** கட்டளை இவ்வாறு பயன்படுத்தப்படுகிறது.

```
COMMIT;
```

12.7.6.2 ROLLBACK கட்டளை

ROLLBACK கட்டளை கடைசி COMMIT கட்டளைக்கு பிறகு கொடுத்த அனைத்து கட்டளைகளையும் திரும்ப பெறும். இந்த கட்டளையுடன் SAVEPOINT கட்டளையும் பயன்படுத்தி ஒரு குறிப்பிட்ட savepoint இடத்திற்கு கொண்டு செல்ல முடியும்.

```
ROLL BACK TO savepoint name;
```

12.7.6.3 SAVEPOINT கட்டளை

SAVEPOINT கட்டளை ஒரு நடவடிக்கையை தற்காலிகமாக சேமித்து வைப்பதால், நாம் தேவைப்படும் சமயங்களில் திரும்பவும் அந்த நிலைக்கு செல்லலாம். நம்முடைய அட்டவணையின் மாறுபட்ட நிலைகளை வெவ்வேறு பெயர்களை பயன்படுத்தி எந்த சமயத்திலும் சேமித்துக் கொள்ளலாம். மேலும், **ROLLBACK** கட்டளையைப் பயன்படுத்தி, மீண்டும் அந்த நிலையை சென்றடையலாம்.

```
SAVEPOINT savepoint_name;
```

பின்வரும் தரவைக் கொண்ட மாணவர் அட்டவணையில் COMMIT, SAVEPOINT மற்றும் ROLLBACK கட்டளைகளை காண்பிக்கும் எடுத்துக்காட்டு:

Admno	Name	Gender	Age	Place
105	Revathi	F	19	Chennai
106	Devika	F	19	Bangalore
103	Ayush	M	18	Delhi
101	Adarsh	M	18	Delhi
104	Abinandh	M	18	Chennai

```
INSERT INTO Student VALUES (107, 'Beena', 'F', 20 , 'Cochin');
COMMIT;
```

Admno	Name	Gender	Age	Place
105	Revathi	F	19	Chennai
106	Devika	F	19	Bangalore

103	Ayush	M	18	Delhi
101	Adarsh	M	18	Delhi
104	Abinandh	M	18	Chennai
107	Beena	F	20	Cochin

UPDATE Student SET Name = 'Mini' WHERE Admno=105;

SAVEPOINT A;

Admno	Name	Gender	Age	Place
105	Mini	F	19	Chennai
106	Devika	F	19	Bangalore
103	Ayush	M	18	Delhi
101	Adarsh	M	18	Delhi
104	Abinandh	M	18	Chennai
107	Beena	F	20	Cochin

INSERT INTO Student VALUES(108, 'Jisha', 'F', 19, 'Delhi');

SAVEPOINT B;

Admno	Name	Gender	Age	Place
105	Mini	F	19	Chennai
106	Devika	F	19	Bangalore
103	Ayush	M	18	Delhi
101	Adarsh	M	18	Delhi
104	vAbinandh	M	18	Chennai
107	Beena	F	20	Cochin
108	Jisha	F	19	Delhi

ROLLBACK TO A;

Admno	Name	Gender	Age	Place
105	Mini	F	19	Chennai
106	Devika	F	19	Bangalore
103	Ayush	M	18	Delhi
101	Adarsh	M	18	Delhi
104	Abinandh	M	18	Chennai
107	Beena	F	20	Cochin

Chapter 12.indd 232 30-01-2020 20:45:17

நினைவில் கொள்க:

- *SQL* என்பது உறவுநிலைத் தரவுத்தளங்களை உருவாக்கவும், செயல்படுத்தவும் உதவும் மொழியாகும்.
- *MySQL* என்பது ஒரு தரவுத்தள மேலாண்மை அமைப்பாகும்.
- *SQL*ன் பல்வேறு கூறுகளானவை தரவு வரையறை மொழி *(DDL),* தரவு கையாளுதல் மொழி *(DML),* தரவு வினவல் மொழி *(DQL),* பரிவர்த்தனைக் கட்டுப்பாட்டு மொழி *(TCL),* தரவு கட்டுப்பாட்டு மொழி *(DCL).*
- *DDL* அட்டவணைகளை உருவாக்குதல் மற்றும் அழித்தலுக்கான கூற்றுகளை வழங்குகிறது.
- *DML* ஒரு அட்டவணையில் தரவை சேர்த்தல், புதுப்பித்தல் மற்றும் நீக்குதலுக்கான கூற்றுகளை வழங்குகிறது.
- *DCL* தரவு அணுகுதலுக்கான அங்கீகார கட்டளைகளை வழங்குகிறது.
- தரவுத்தளத்திலுள்ள பரிவர்த்தனைகளை மேலாண்மை செய்ய *TCL* கட்டளைகள் பயன்படுகின்றன.
- *DQL* கட்டளைகள் தரவுத்தளத்தில் வினவல்கள் உருவாக்க உதவுகின்றன.
- *CREATE TABLE* கட்டளை ஒரு புதிய அட்டவணையை உருவாக்குகிறது.
- *SELECT* கட்டளை வினவல்களை உருவாக்க உதவுகிறது. மேலும், அதனை **WHERE clause** பயன்படுத்தி பல்வேறு வகைகளில் கட்டுப்படுத்த முடியும்..
- *INSERT* கட்டளை அட்டவணையில் வரிசைகளை சேர்க்க அனுமதிக்கிறது.
- *DELETE* கட்டளை அட்டவணையில் வரிசைகளை நீக்க அனுமதிக்கிறது.
- *UPDATE* கட்டளை ஏற்கனவே உள்ள வரிசைகளின் மதிப்புகளுக்கு மாற்றங்கள் செய்ய அனுமதிக்கின்றது.
- *ALERT TABLE* கட்டளை அட்டவணையின் வடிவமைப்பை மாற்றியமைக்க அனுமதிக்கிறது.
- *TRUNCATE* கட்டளை அனைத்து பதிவுகள் மற்றும் அட்டவணையை நீக்கி, ஒதுக்கீடு செய்த இடத்தை விடுவிக்கப் பயன்படுகிறது.
- *DROP TABLE* கட்டளை தரவுத்தளத்திலிருந்து ஒரு அட்டவணையை நீக்குகிறது.
- *COMMIT* கட்டளை எந்தவொரு பரிவர்த்தனையையும் தரவுத்தளத்தில் நிரந்தரமாக சேமித்து வைக்கிறது.
- *SAVEPOINT* கட்டளை தேவையானப் போது, அந்த குறிப்பிட்ட இடத்திற்கு ஒரு குறிப்பிட்ட இடத்தில் பரிவர்த்தனையை தற்காலிகமாக சேமிக்கிறது.
- *ROLLBACK* தரவுத்தளத்தை அதன் கடைசி *COMMIT* கட்டளைக்கு பிறகு கொடுத்த அனைத்து கட்டளைகளையும் திரும்ப பெறும்.

செய்முறைப் பயிற்சி

1. பணியாளர் எண், பணியாளர் பெயர், பதவி, பணியில் சேர்ந்த தேதி, அடிப்படை ஊதியம் ஆகிய புலங்களைக் கொண்ட ஒரு பணியாளர் அட்டவணையை உருவாக்குக.
2. மேற்கூறிய அட்டவணையில் பணியாளர் எண்ணை முதன்மை திறவுகோல் (primary key) ஏற்படுத்திக் கொள்ளவும். மேலும், ஏதேனும் ஒரு புலம் NOT NULL-லாக குறிப்பிடுக.
3. மேலாளர்களாக இருக்கும் அனைத்து பணியாளர்களின் பட்டியலை தயாரிக்கவும்.

செய்முறைப் பயிற்சி

1. மாணவர் அட்டவணைக்கு **name, age, place** மற்றும் **admno** என்ற புல வரிசைப்படி வினவலை உருவாக்கு.
2. மாணவர் அட்டவணையில் **18** வயதுக்கு மேற்பட்ட மாணவர் பற்றிய விவரங்களை வெளியிட ஒரு வினவலை உருவாக்கவும். **City** என்ற புலத்திற்கு **unique** கட்டுப்பாட்டை வரையறுத்துக் கொள்ளவும்.

F93S9E

மதிப்பீடு

பகுதி–அ

சரியான விடையை தேர்ந்தெடுத்து எழுதுக **(1 மதிப்பெண்)**

1. எந்த கட்டளைகள் அட்டவணை வடிவமைப்பை உருவாக்குதல், உறவுநிலையை நீக்குதல் மற்றும் உறவுநிலை திட்ட வடிவமைப்பை மாற்றுதல் போன்ற செயற்பாடுகளுக்கான வரையறைகளை வழங்குகிறது?

   அ) DDL ஆ) DML

   இ) DCL ஈ) DQL

2. எந்த கட்டளை அட்டவணையின் வடிவமைப்பை மாற்றி அமைக்க அனுமதிக்கிறது?

   அ) SELECT ஆ) ORDER BY

   இ) MODIFY ஈ) ALTER

3. அட்டவணையை நீக்க பயன்படுத்த வேண்டிய கட்டளை

   அ) DROP ஆ) DELETE

   இ) DELETES ALL ஈ) ALTER TABLE

4. வினவல்களை உருவாக்க பயன்படுவது

   அ) SELECT ஆ) ORDER BY

   இ) MODIFY ஈ) ALTER

5. ஒரு தரவுத்தளத்தில் உள்ள தரவை வரிசையாக்கம் செய்ய பயன்படும் clause

   அ) SORT BY ஆ) ORDER BY

   இ) GROUP BY ஈ) SELECT

பகுதி–ஆ

அனைத்து வினாக்களுக்கும் விடையளி **(2 மதிப்பெண்)**

1. 18 வயதிற்கும் குறைவாக உள்ள அனைத்து மாணவர்களின் தரவினை வரிசைப்படி தெரிவு செய்யும் ஒரு வினவலை எழுதுக.
2. Unique மற்றும் Primary Key கட்டுபாடுகளை வேறுபடுத்துக
3. அட்டவணை கட்டுப்பாட்டிற்கும், நெடுவரிசை கட்டுப்பாட்டிற்கும் உள்ள வேறுபாடுகளை எழுதுக?

Chapter 12.indd 234 30-01-2020 20:45:18

4. எந்த SQL கூறு, அட்டவணையை உருவாக்கவும், அவற்றில் மதிப்புகளை சேர்க்கவும் அனுமதிக்கும்?
5. SQL மற்றும் MySQLக்கு உள்ள வேறுபாடுகள் யாவை?

பகுதி–இ

அனைத்து வினாக்களுக்கும் விடையளி **(3 மதிப்பெண்)**

1. கட்டுப்பாடு என்றால் என்ன? Primary Key கட்டுப்பாடு பற்றி சிறுகுறிப்பு வரைக.
2. ஒரு புதிய புலத்தை சேர்ப்பதன் மூலம் மாணவர் அட்டவணை கட்டமைப்பை மாற்றி அமைக்க ஒரு SQL கூற்றை எழுதுக.
3. ஏதேனும் மூன்று DDL கட்டளைகளை எழுதுக.
4. Savepoint கட்டளையின் பயன்பாட்டை ஒரு எடுத்துக்காட்டுடன் எழுதுக.
5. DISTINCT சிறப்புச் சொல்லை பயன்படுத்தி ஒரு SQL கூற்றினை எழுதுக.

பகுதி–ஈ

அனைத்து வினாக்களுக்கும் விடையளி **(5 மதிப்பெண்)**

1. பல்வேறு வகையான கட்டுப்பாடுகளையும், அதன் செயல்பாடுகளையும் எழுதுக.
2. கீழ்காணும் பணியாளர் அட்டவணையை கருத்தில் கொண்டு, (i) முதல் (v) வரையிலான வினாக்களுக்கு SQL கட்டளைகளை எழுதுக.

EMP CODE	NAME	DESIG	PAY	ALLO WANCE
S1001	Hariharan	Supervisor	29000	12000
P1002	Shaji	Operator	10000	5500
P1003	Prasad	Operator	12000	6500
C1004	Manjima	Clerk	8000	4500
M1005	Ratheesh	Mechanic	20000	7000

(i) அனைத்து பணியாளர்களின் விவரங்களை அவர்கள் பெறும் சம்பளங்களின் இறங்கு வரிசையில் காண்பிக்க.

(ii) 5000 முதல் 7000 வரை ALLOWANCE பெறும் அனைத்து பணியாளர்களின் விவரங்களை காண்பிக்க.

(iii) mechanic வகையை சார்ந்த பணியாளர்களை நீக்க.

(iv) ஒரு புதிய வரிசையை உருவாக்க.

(v) operators வகையை சார்ந்த அனைத்து பணியாளர்களின் விவரங்களை காண்பிக்க.

3. SQLன் கூறுகள்? ஒவ்வொன்றிற்கும் கட்டளைகளை எழுதுக.
4. மாணவர் அட்டவணையில் பின்வரும் SQL கூற்றுகளை கட்டமைக்கவும்.

   (i) SELECT கூற்று GROUP BY clause பயன்படுத்தி

   (ii) SELECT கூற்று ORDER BY clause பயன்படுத்தி
5. பணியாளர்களுக்கான ஏதேனும் ஐந்து புலங்களைக் கொண்ட ஒரு அட்டவணையை உருவாக்க ஒரு SQL கூற்றினை எழுதி, அந்த பணியாளர் அட்டவணைக்கு ஒரு அட்டவணை கட்டுப்பாட்டை உருவாக்கவும்.

Chapter 12.indd 235 30-01-2020 20:45:18