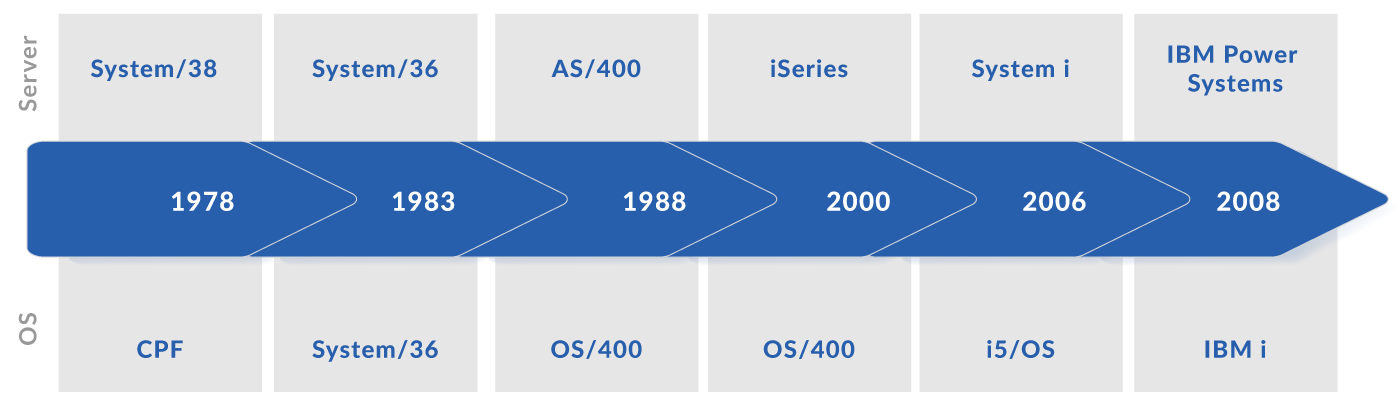
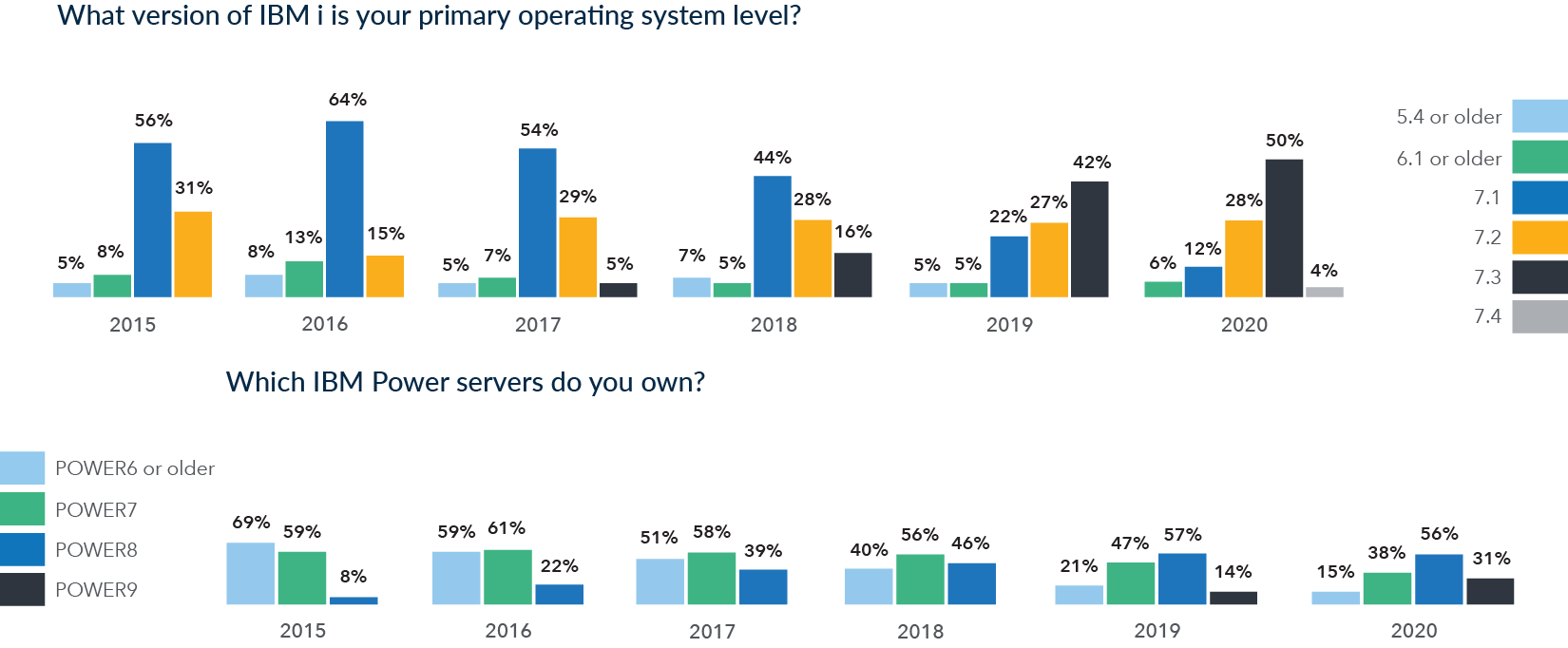
**Introduction to AS400**

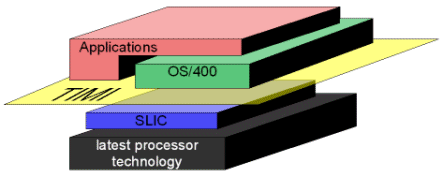
* The Application System/400 (AS/400) is an IBM minicomputer for general business and departmental use.
* Introduced in 1988 and still in production under the names iSeries.
* The AS/400 is an object oriented system with an integrated database.



* IBM i V5R4 was released in Feb 2006 and supported till Sep 2013
* IBM i 6.1 was released in Mar 2008 and supported till Sep 2015
* IBM i 7.1 was released in Apr 2010 and supported till Apr 2018
* IBM i 7.2 was released in Nov 2014 and supported till Apr 2021
* IBM i 7.3 was released in Apr 2016 and still supported
* IBM i 7.4 was released in Jun 2019 and still supported
* To see AS400 version on system, Type command DSPSFWRSC, press enter and press F11
* DSPJOB OUTPUT(\*PRINT) and press ENTER



System Architecture



* Languages
  + RPG/400
  + COBOL/400
  + AS/400 BASIC
  + AS/400 PASCAL
  + STRUCTURED QUERY LANGUAGE
  + C/400
  + FORTRAN/400
  + Java/400
  + C++/400

Single Level Storage

* OS/400 applications and the objects with which they interact all reside in a very large virtualized, single-level storage.
* Storage management, a component of SLIC, ensures that objects which need to persist when the system is off are maintained in persistent storage.
* Advantages
  + Applications need not tailor their memory usage to a specific machine configuration.
  + Single Level Storage greatly facilitates the on-demand allocation of memory among logical partitions.

**AS400 Work Management**

* Describes where work enters system
* Where and with what resources work is processed
* Where the output goes
* Supports the commands and internal functions
* Work comes from
  + job queues, workstation, autostart jobs, prestart jobs, communication jobs, interactive
* Where and with what resources work is processed
  + Memory pool allows to control storage of each subsystem
  + Routing Entries guides job to its correct place
  + Class object manage priority, storage, cpu time of each job
* How work leaves the system
  + The output queue like job queue schedules output to be printed. Both printer output and outq carry attributes to print the info
* AS400 Work management terms
  + OS/400
  + Subsystem
  + Job
  + Library
  + Objects

OS/400

* All AS400 models are supported by a single, integrated os called operating system 400(os/400)
* The key components/functions of OS/400 include

Work Management and controls job processing – This allows multiple interactive and batch jobs to run simultaneously in different subsystems and job queues. It also includes jobs and governs input and output spool functions.

Database Management – Facilitates retrieval, addition and updating of data on the AS400 database. It adopts the relational model of DBMS. This allows users to use pf and lf. It also permits data definitions to be independent from application programs, thus ensures greater data integrity

Communications Support – Offers a wide range of communications and networking capabilities. It also AS400 to communicate and transfer data among AS400 machines, PC’s

Control Language provides CL commands to perform system, operational and programming functions.

Application Development Support provides utilities such as SEU, DFU, PDM, SDA, RLU

Message handler controls communications between users, between users and operating system, between users and programs

System Security policies system access, manages object authorizations, protects data and system resources from unauthorized access

Query Management provides a query utility that enables users to retrieve data from the database and to create/format reports.

Object Management provides functions to create objects, to maintain objects and to locate/retrieve objects for processing

Storage Management performs functions necessary for placing objects into storage, retrieving them from storage.

Subsystems

* A specialised environment for handling a certain type of work or function, such as
  + Interactive Processing
  + Batch Processing
  + Spooling
* AS400 has several subsystems, each working independently and performing specific task
* Job is assigned to a subsystem based on type of job
* Subsystem description defines memory pools allocated, routing entry, max active jobs, no of active jobs in subsystem. Object type is \*SBSD
* Types of Subsystem
  + QBASE – Supports Interactive, Batch and communication jobs
  + QBATCH – Supports Batch jobs
  + QINTER – Supports Interactive jobs
  + QCMN – Supports all communication jobs
  + QSPL – Spool subsystem that supports reader/writer jobs
  + QCTL- Controlling subsystem that starts up system console
* Steps to create subsystems
  + CRTSBSD to create subsystem description
  + CRTJOBQ to create job queue
  + ADDJOBQE to add entries to the Job Queue
  + CRTCLS to create class to define attributes like run priority,

    time slice, default wait time, max temporary storage.

* ADDRTGE to add routing entry
* STRSBS to start subsystem
* Once the subsystem is started, it can be used to submit jobs
* ENDSBS on a subsystem to end it

Job

* A job can enter subsystem via job queue entry, workstation, autostart job entry, prestart job entry
* WRKJOB command followed by job name to see attributes of a job
* WRKUSRJOB command followed by username to see all the jobs of a user and their attributes
* WRKACTJOB to see all active jobs
* Above 3 commands can be used to hold/release/end job/change job attributes
* Job Types
  + Autostart job – Is a batch job, doing repititve and one time initialization work that is associated with a particular subsystem
  + Batch job – Predefined processing actions submitted to the system to be performed with little or no interaction between the user and the system. Batch jobs can run in system background, freeing the user who has submitted it to do other work. Several batch jobs can be active at the same time.
  + Interactive job – Interactive job is a job that starts when a user signs on to AS400 system and ends when the user signs off.
  + Prestart jobs – Is a batch job that starts running before a work request is received.
  + Communication jobs – Is a batch job that is started by a program start request from a remote system.  e.g. FTP, SNDNETF command is used for communication.
* Autostart job
  + Is a batch job
  + Starts with associated subsystem or using STRAJ command
  + Performs initialization and repetitive work of a subsystem.
* Prestart job
  + Is a batch job
  + Starts running before a work request
  + Starts with subsystem or STRPJ command
  + Use prestart job entries in subsystems
  + Determines program, storage pool to use when jobs are started
* Interactive job
  + Interactive jobs are started when the user signs on to a workstation and ends on signing off.
* Batch job
  + Predefined processing actions submitted to the system to be performed with little or no interaction between the user and the system.
  + These jobs do not require human interaction
  + Several batch jobs can be active at the same time
* Job Status
  + ACTIVE – job has started
  + ACTIVE HELD – job has started and is held
  + OUTQ – job in output queue has completed(successfully/unsuccessfully) and has spool files in output queue
  + JOBQ – job is in Job queue
  + JOBQ HELD – job is in Job queue and is held
  + OUTQ HELD - job in output queue has completed(successfully/unsuccessfully) and has spool files in output queue which are held
  + SCD – Job is scheduled for a date/time.
  + MSGW – Job is waiting for a interactive reply from a message.
* Job Characteristics/Attributes
  + Job Name – For interactive jobs, name is same as workstation/user id login. For batch jobs, you specify the user profile upto 10 character long.
  + Job Number – Unique 6 digit number assigned by the system.
  + Same job names will have different job number
  + Job Description – Tells the system when to start the job, where to get job from, how job will run.
  + Job descriptions cannot be overridden for autostart, workstation, communication jobs.
  + A job description that has a USER specified should be authorized only to specific individuals
  + Eg SBMJOB JOB(DAILY) JOBD(QBATCH) USER(XYZ)
  + Run Priority – 2 digit(00-99) number specifies the priority level assigned to all jobs running.
* Job Scheduling
  + Can schedule jobs to be released from the job queue at a particular time
  + IBM Advanced Job Scheduler for licensed program is a powerful scheduler that allows unattended job processing 24hrs/day 7days/week
  + ADDJOBSCDE command is used for adding job schedule entry

ADDJOBSCDE JOB(MONTHEND) CMD(CALL INVENTORY)

SCDDATE(\*MONTHEND) SCDTIME(‘23:30:00’) FRQ(\*MONTHLY)

OMITDATE(‘12/31/05)

This will submit a job to run program INVENTORY at 11:30pm n last day of every month except on New Year’s Eve.

* Job Queue
  + Job Queue contains ordered list of jobs waiting to be processed by a subsystem
  + Job queue is first place where submitted job goes before becoming active in a subsystem.
  + There must be an active subsystem accepting work from job queue for jobs of job queue to be processed
  + CRTJOBQ to create a job queue
  + DLTJOBQ to delete a job queue. The job queue being deleted cannot contain any entries. All jobs on queue must be completed, deleted or moved to different job queue. The subsystem cannot be active to the job queue.
* Output Queue
  + Output Queues are objects where printer output files(spooled files) wait to be processed and sent to printer
  + Printer output is created by system or user using a print file.
  + If a specified output queue cannot be found, the printer output will be directed to QPRINT in library QGPL.
* Spooled files
  + Spooling saves data for later processing or printing.This data is stored in spooled file.
  + Spooled files allow to manage data for externally attached devices such as a printer.
  + Two types of spooling. Output spooling can be used for printer devices and diskette devices. Input spooling for database file output
  + Batch and Interactive job processing can result in spooled output records that are to be processed on an output device such as printer or diskette device. These output records are stored in spooled files until processed.
  + A single job can have many spooled files.
  + A job can have spooled files on more than output queues.
  + WRKSPLF command to work with all the spool files for current user
  + Can hold/release/change/delete/display/SNDNET spool file
  + WRKSPLF SELECT(\*CURRENT \*ALL \*ALL \*ALL \*ALL
  + XYZ)
* Job Log
  + Contains info related to requests entered for a job.
* DSPJOBLOG command to see the job log of a user/job
* Log parameter has 3 attributes

Message logging level determines which messages and what message types should be logged for the job

Message Severity level controls minimum message severity to

        log(00-99, 00 – informational, 99 – most severe)

Message text level that specifies level of detail in job log

* LOG attribute can be set in commands

CHGJOBD ….LOG(4 0 \*SECLVL)

SBMJOB    ….LOG(4 30 \*SECLVL) – for batch job

CHGJOB    ….LOG(4 0 \*NOLIST) – for active job

* The different message logging level and if messages are kept in job log

0 – No messages to keep in job log

1 – Only messages sent to \*EXT message queue

2 – Messages sent to \*EXT message queue, logged commands that issue messages, messages generated by logged commands

3 – Messages sent to \*EXT message queue, all logged commands, messages generated by logged commands

4 – All messages, all logged commands

* The different message text levels are

\*NOLIST – If job ends normally, job log is not spooled

\*MSG – Only first level message is logged, job log is spooled whether job ends normally or abnormally

\*SECLVL – First and second level(detailed) message text is logged, Job Log is spooled whether job ends normally or abnormally

DSPMSG

* The DSPMSG(display Message) command is used by the display station user to show the messages received at the specified message queue.
* Message Queue(MSGQ)
* Specifies the message queue from which messages are shown
* The special values of \*WRKSTN, \*WRKUSR, \*USRPRF, \*SYSOPR should be specified for this parameter only when you are in an interactive job.
  + For \*SYSOPR, messages from QSYSOPR are shown
  + For \*WRKSTN, messages from work station’s own message queue are shown
  + For \*USRPRF, messages from current user profile message queue is shown.
* Message Type

Specifies the type of messages in the message queue

\*ALL – all messages are shown in message queue

\*INFO – only informational messages are shown

\*INQ – only inquiry messages are shown

\*COPY – copies of inquiry messages that were sent to other message queue and still require reply are shown

* Error Messages
* These are messages which can result due to error in AS400 command parameter input values
* Object type is \*ESCAPE
* Escape message id starts with CPF. Eg CPFNNNN where NNNN is numeric
* Eg  CPF2203 – User profile &1 not correct
* CPF2217 – Not authorized to user profile &1
* CPF2403 – Message queue &1 in &2 not found
* QUSRMSG is system message queue and WRKMSGQ is the command to access message queue/messages.

User Profile

* Object Type of \*USRPRF
* Menu options for user profile is 9->20->38
* User profiles shipped with AS/400

**Customer Use Service Personnel Use**

QSECOFR QSRV         - full

QPGMR QSRVBAS  - basic

QSYSOPR QTSTRQS  - test request

QUSER

* CRTUSRPRF to create an user profile, CHGUSRPRF to change an user profile, WRKUSRPRF to change/copy/delete/display an user profile

Library

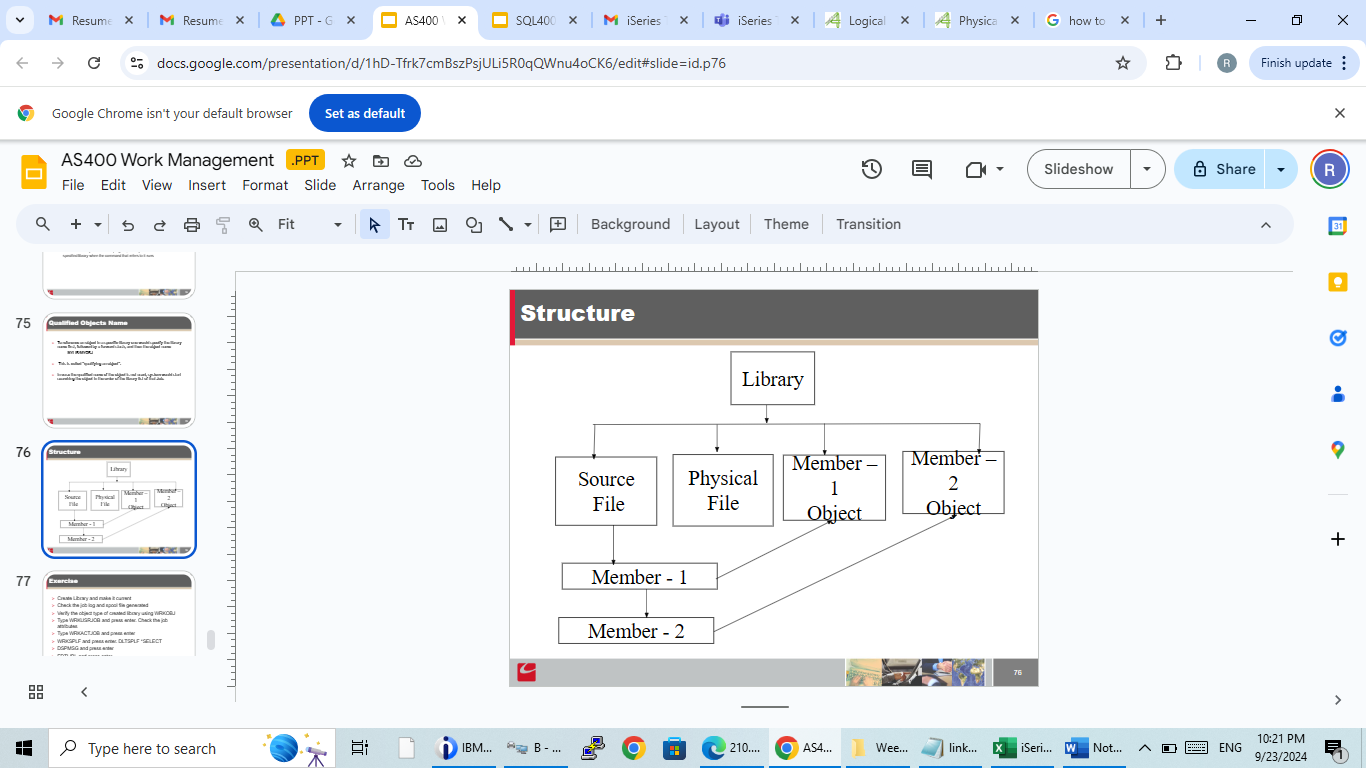
* A library is an object used to group related objects and to find objects by name
* Thus, a library is a directory to a group of objects
* When command/program is executed, AS400 determines the command/program location by use of libraries.
* Library is a Collection of Objects. QSYS is the only library that contains other library.
* Library has an Object type of \*LIB.
* Types of Libraries
  + PROD – Added automatically and removed when job completes
  + USER – Libraries created by user
  + SYS – All IBM supplied libraries
  + CUR – Current library is the working library
* PROD(Product) libraries
  + Two product libraries may be included in the library list
  + The system uses product libraries to support languages and utilities that are dependent on libraries other than QSYS to process their commands
* Current library
  + The current library can be, but does not have to be, a duplicate of any library in the library list
  + The value \*CURLIB (current library) may be used on most commands as a library name to represent whatever library has been specified as the current library for the job
  + You can change the current library for a job by using the Change Current Library (CHGCURLIB) or Change Library List (CHGLIBL) command
* System library
  + The system part of the library list contains objects needed by the system
  + System libraries are established when the operating system is installed.
* **User libraries**
  + User libraries are the libraries created by individual users.
  + Each user can have more than one user library. However, there can be only one current or default library, called the current library, for each user.
* **QSYS**

**System Objects  
SLIC / LIC  
User Libraries**

* **QTEMP   
  Temporary Library created for each job.**
* **QGPL   
  General purpose library**
* Authority for Libraries
  + The types of combined authority include \*USE authority, \*CHANGE authority, \*ALL authority, and \*EXCLUDE authority
* Below are widely used AS400 object commands.
  + Lock states for objects
  + A lock state identifies the use of the object and whether it is shared
  + Objects are allocated on the basis of their intended use (read or update) and whether they can be shared (used by more than one job)
  + The file and member are always allocated \*SHRRD and the file data is allocated with the level of lock specified with the lock state
  + The five lock states are (parameter values given in parentheses):
  + **Exclusive (\*EXCL)**
  + The object is reserved for the exclusive use of the requesting job; no other jobs can use the object
  + **Exclusive allow read (\*EXCLRD)**
  + The object is allocated to the job that requested it, but other jobs can read the object
  + **Shared for update (\*SHRUPD)**
  + The object can be shared either for update or read with another job
  + **Shared for read (\*SHRRD)**
  + The object can be shared with another job if the user does not request exclusive use of the object
  + The Work with Object Locks (WRKOBJLCK) command or the Work with Job (WRKJOB) command displays the lock states for objects
  + To reference an object in a specific library one would specify the library name first, followed by a forward slash, and then the object name

MYLIB/MYOBJ

This is called "qualifying an object”.



**Files**

* AS/400 uses the integrated database DB2/400 as a Relational Database.
* All RDBMS rules are applicable to DB2/400
* AS/400 files are broadly of 4 types

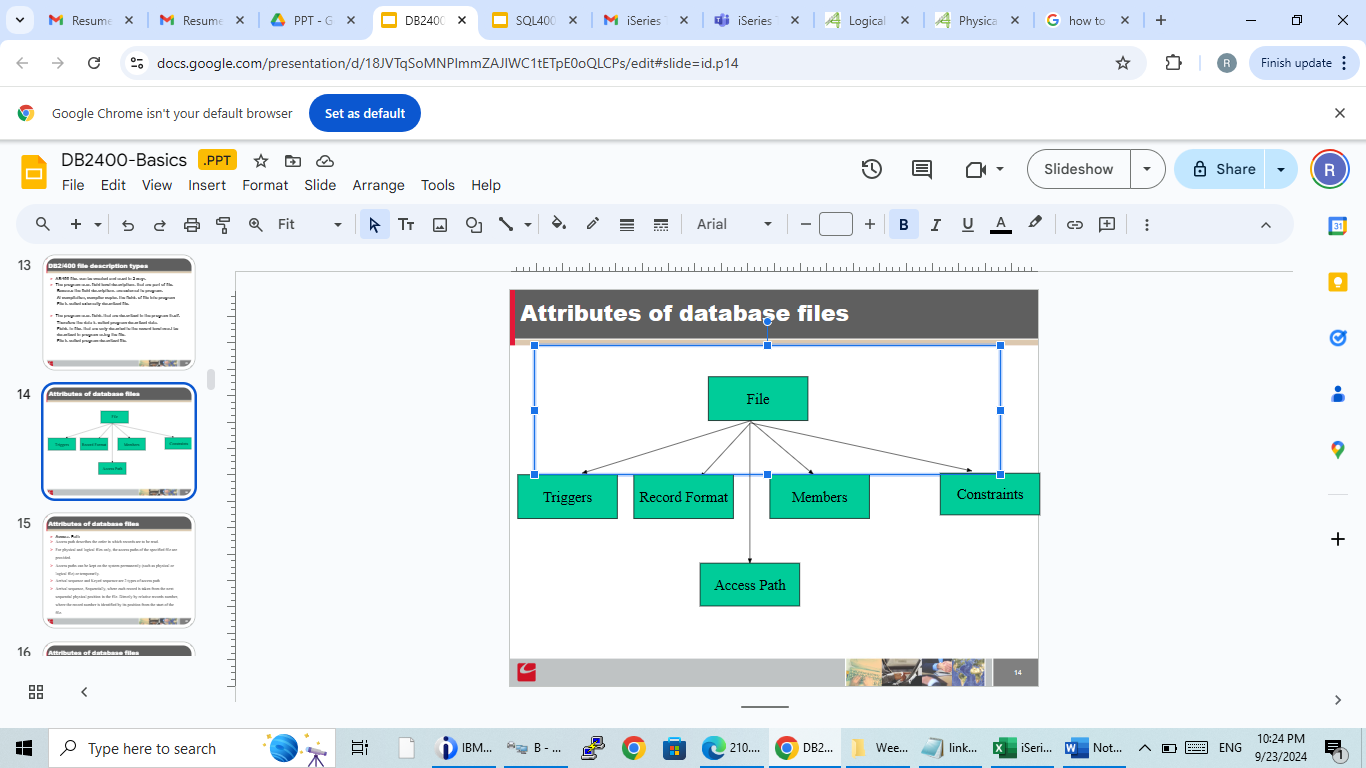
Database files

Device files

Save files

DDM files

Tape files

* DSPFD command to check the attributes of a file
* Object type is \*FILE
* **Device files**
  + Contain no actual data
  + Contain information valid for device type the application is processing
  + Types of device files are display, printer, tape, diskette, ICF
  + Display files, ATTR(DSPF) provide specific information to how an application can interact with a workstation. Contains no data. Has record format
  + Printer files, ATTR(PRTF) provide information to how an application can spool data for output to a writer.
* **DDM files**
  + Distributed Data management files, ATTR(DDMF) are objects that represent files that exist in a remote system.
* **SAVE files**
  + ATTR(SAVF), are a special file designed specially to handle save/restore data.
* 
* **Triggers**
  + A trigger defines a program that is called with a delete, insert, update operation occurs for a file.
  + The trigger program can be specified to be called before or after a change operation occurs. The change operation can be an insert, update, delete or read operation through any interface.
  + A maximum of 300 triggers can be added to one physical file. The trigger program to be called can be the same for each trigger or it can be a different program for each trigger.
  + ADDPFTRG and RMVPFTRG commands to add and remove triggers from pf.
* **Constraints**
  + A constraint is a rule that is used for optimization purposes.
  + The four types of constraint relationships that you can add are referential constraints, unique constraints, primary key constraints and check constraints.
  + All constraints are defined at the file level.
  + You can use constraint relationships to define dependencies between files.
  + ADDPFCST command to add constraint to a file.
  + \*REFCST is for referential constraint, \*UNQCST for unique constraints, \*PRIKEY for primary key constraints, \*CHKCST for check constraints

**Physical Files**

* A physical file is a database file that stores application data
* The maximum number of fields in a record format is 8000
* CRTPF to create a physical file object. This will also delete all the existing data in pf
* Valid Data Types are

A Character

P Packed decimal

S Zoned decimal

B Binary

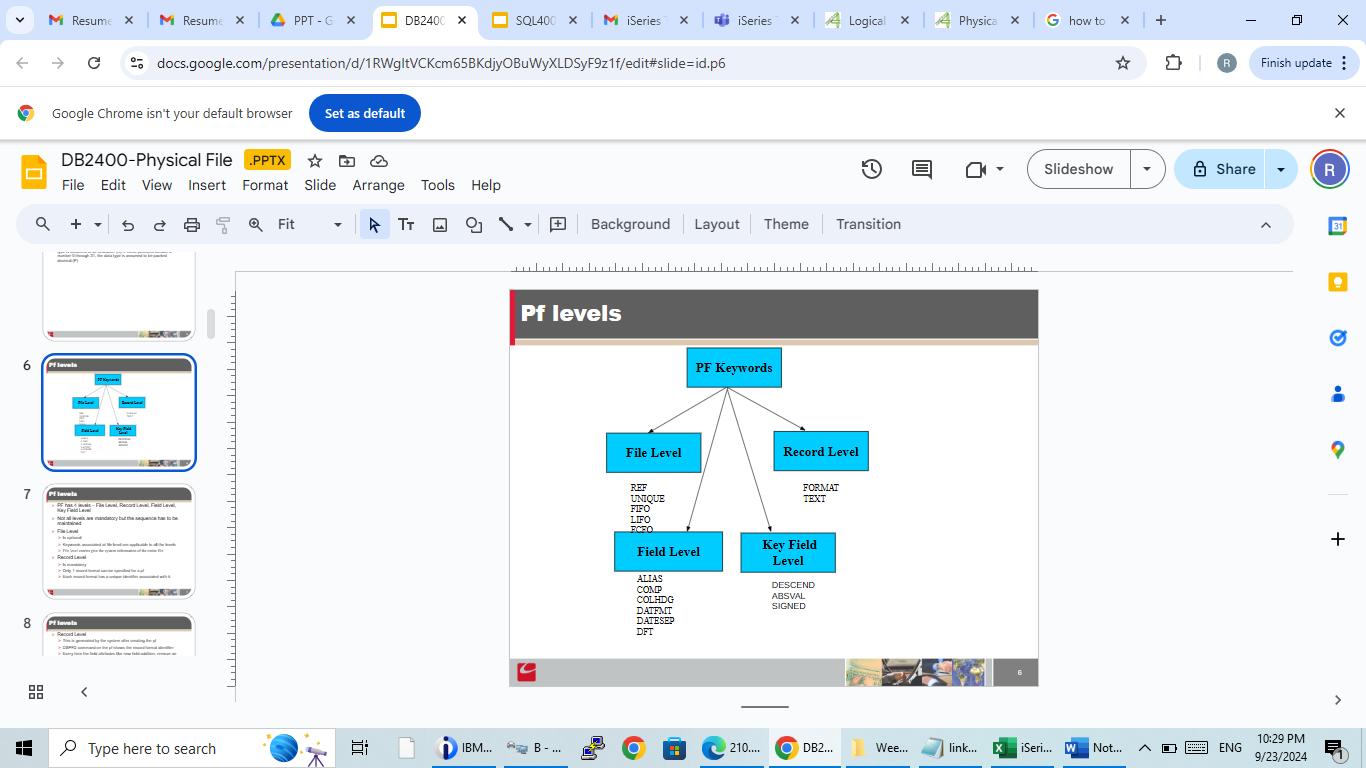
F Binary floating point

H Hexadecimal

L Date

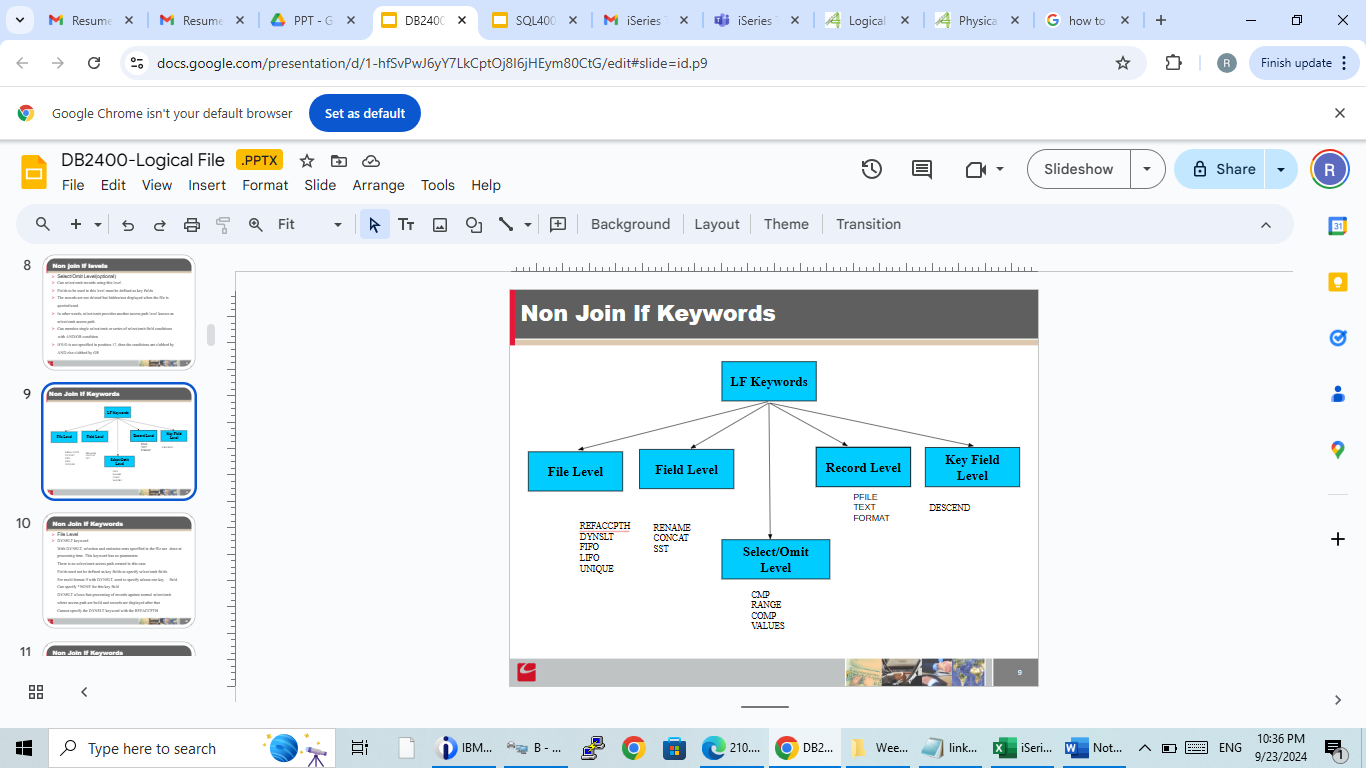
T Time

Z Timestamp



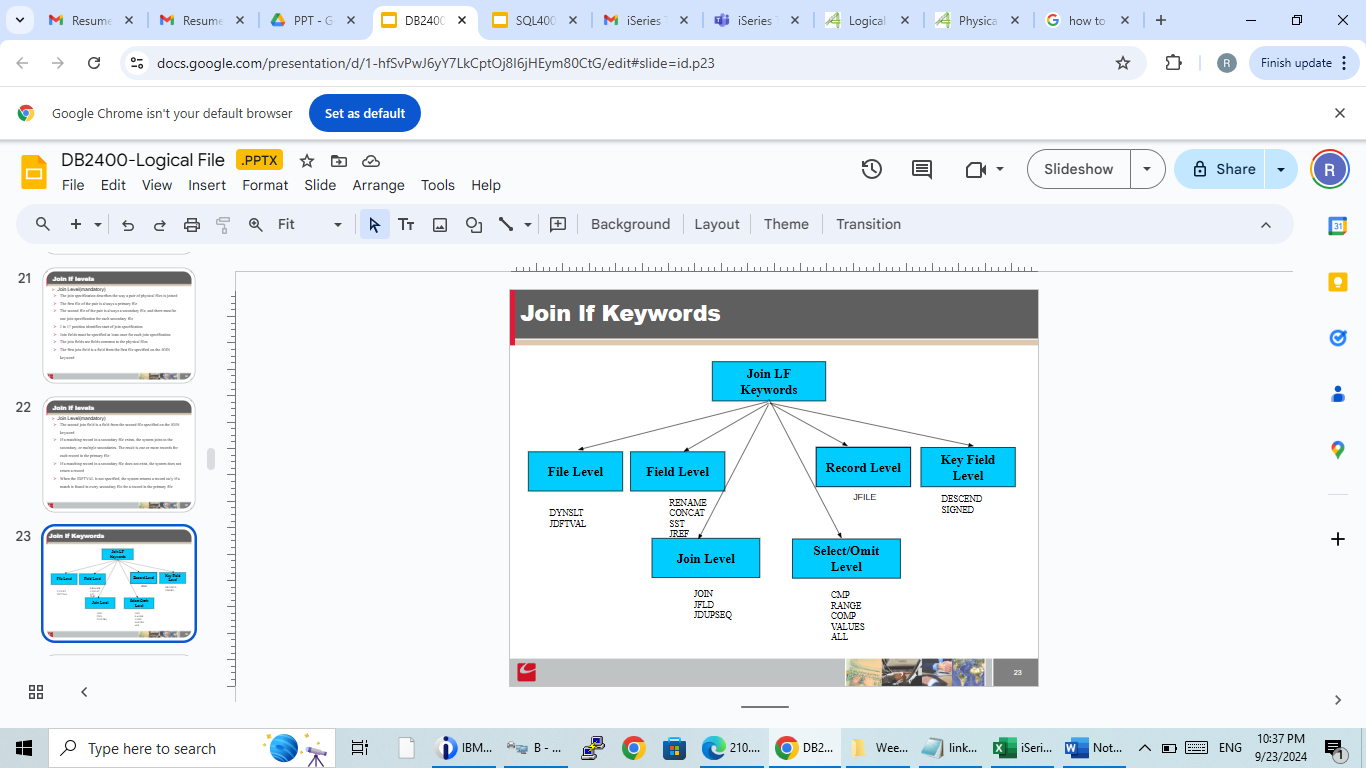
**Logical Files**

* A logical file does not contain any data but provides the ‘VIEWS’ of the data to satisfy end-user’s needs
* A logical file can contain up to 32 record formats
* It selects records dynamically. It cannot exist without a physical file
* PFILE keyword in the Record Level is mandatory.
* There are 2 types of lf, non-join lf and join lf
* Non join lf does not join any pf
* Join lf joins 2 or more(till 32) pf
* CRTLF is the command to create lf
* Non join lf types are of single format and multi format
* Single format lf
* Is derived from single pf
* Multi format lf
* Has 2 or more record format.
* Each record format of such a multiple-format logical file is always associated with one or more physical files.
* Can use the same physical file in more than one record format.



Join LF

* Is a logical file that combines two or more PF.
* A PF cannot be changed through a JLF.
* DFU cannot be used to display a JLF.
* Only one record format can be specified in a JLF.
* Commitment control cannot be used with a JLF.
* Max 32 files can be joined.
* At a time only 2 files can be joined. The first file is primary and other secondary.
* Join lf types are of natural join, left outer join, self join
* Self join is joining same file
* Natural join will return result rows if both files contain a column name with same name and data type and value
* Left outer is joining 2 different files, primary and secondary



**STRDFU**

* STRDFU(Start Data File Utility) is an IBM tool for defining and manipulating data in a relational database
* DFU performs typical DML functions
* DML(Data Manipulation Language) allows to add/modify/delete data
* If file is empty, default is entry mode
* If file has data, default is change mode
* F9 – Insert Mode
* F10 – Entry Mode
* F11 – Change Mode
* F23 – Deletes current record
* Page Down to keep scrolling down for next records
* Page Up to keep scrolling up for previous records
* If file has key fields, records can be searched by entering key field values
* If file has no key fields, records can be searched by entering record no/rrn
* You cannot edit rrn/key values in change mode

**Query/400**

Query /400 is an IBM licensed program and a decision support

utility that can be used to obtain information from the AS/400

database(DB2/400)

It can obtain information from any database files that have been

defined on the system

Query/400 definition object type is \*QRYDFN

Query/400 can be used to Select, arrange and analyze data stored

     in one or more database files (Maximum up to 32) to produce

     reports and other data files

 New query definitions can be created  by selecting all the fields from

      a file or selecting few fields from a database file

Commands

* WRKQRY (Work with Queries)
* STRQRY (Start Query)
* GO QUERY
* GO CMDQRY
* RUNQRY (Run Query)
* DLTQRY (Delete Query)
* Query output types
  + Display
  + Printer
  + Database file
* Types of Join
* 1= Matched records

Selects only the records that have matching records in  all the joined files.

This type of join uses only records from each file that has a match with at

 least one record in each and every one of the other selected files.

* 2= Matched records with primary file
* Selects every record in one file (the primary file), and includes all the

 matching records from all the other (secondary) files.  Every record in the

    primary file is selected whether or not it has a match.  (The primary file is

 always the first file specified in your query definition.)

* 3= Unmatched records with primary file

     Selects, from the primary file, only records that have no match in at least

          one of the secondary files. That is, every primary record is selected that

   does not have a matching record in all the secondary files.

QM Query

* Start Query Management Tool using CL command STRQM.
* Select option ‘1’  to Work with QM Queries
* Specify the library name and option to create/change/display a query.
* Select options and define the query on prompt
* Using ‘F3’ key to save and exit the prompted query.
* Convert Query type to SQL using option ’10’.
* QM Query can also be created by exporting a Flat file with SQL statements using CRTQMQRY command.
* Use option ‘9’ to run a query.
* Can be executed through program using STRQMQRY command.

**SQL400**

* Structured Query Language (SQL) is a standardized language for defining and manipulating data in a relational database
* There are several basic types of SQL statements. They are listed here according to their functions.
* SQL schema statements, also known as data definition language (DDL) statements
* SQL data and data change statements, also known as data manipulation language (DML) statements
* Dynamic SQL statements
* Embedded SQL host language statements
* Interactive sql using STRSQL command
* Run SQL Statements(RUNSQLSTM) CL command

This command is used to run SQL statement from Command Line or CL program

* Data Definition Language (DDL) Statements
  + ALTER TABLE
  + CREATE ALIAS
  + CREATE INDEX
  + CREATE PROCEDURE
  + CREATE TABLE
  + CREATE VIEW
  + DROP ALIAS
  + DROP INDEX
  + DROP PROCEDURE
  + DROP TABLE
  + DROP VIEW
  + GRANT PROCEDURE
  + GRANT TABLE
  + RENAME
  + REVOKE PROCEDURE
  + REVOKE TABLE
* Data manipulation language (DML) describes the portion of SQL that manipulates or controls data
* DML Statements are
  + CLOSE
  + COMMIT
  + DECLARE CURSOR
  + DELETE
  + FETCH
  + INSERT
  + LOCK TABLE
  + OPEN
  + ROLLBACK
  + SELECT INTO
  + SET variable
  + UPDATE
  + VALUES INTO
* SELECT Statement
* Select statement looks like this

SELECT column names

FROM table or view name

WHERE search condition

GROUP BY column names

HAVING search condition

ORDER BY column-name

You can specify one or more columns or expressions in the GROUP BY clause to group the rows.

When GROUP BY is used, the function is applied to each group, thereby returning as many rows as there are groups.

Eg. SELECT WORKDEPT, DECIMAL (AVG(SALARY),5,0) FROM CORPDATA.EMPLOYEE GROUP BY WORKDEPT

BETWEEN/AND

Is used to specify a search condition that is satisfied by any value that falls on or between two other values

To find all employees who were hired in 1987, you can use this: ...

WHERE HIREDATE BETWEEN ’1987-01-01’ AND ’1987-12-31’

The BETWEEN keyword is inclusive

IN

IN says you are interested in rows in which the value of the specified expression is among the values you listed

For example, to find the names of all employees in departments A00, C01, and E21, you can specify: ...

WHERE WORKDEPT IN (’A00’, ’C01’, ’E21’)

EXISTS

EXISTS says you are interested in testing for the existence of certain rows.

For example, to find out if there are any employees that have a salary greater than 60000, you can specify: EXISTS (SELECT \* FROM EMPLOYEE **WHERE SALARY > 60000)**

%

A percent sign stands for an unknown string of 0 or more characters

If the percent sign starts the search string, then SQL allows 0 or more character(s) to precede the matching value in the column

For example, to find out which employees live in Minneapolis, you can specify: ...

WHERE ADDRESS LIKE ’%MINNEAPOLIS%’

Join Table

Default join is inner join. INNER JOIN keyword is optional.

SELECT EMPNO, LASTNAME, PROJNO FROM CORPDATA.EMPLOYEE INNER JOIN CORPDATA.PROJECT ON EMPNO = RESPEMP WHERE LASTNAME > ’S’

USING Clause

The USING clause is equivalent to a join condition where each column from the left table is compared to a column with the same name in the right table.

For example, look at the USING clause in this statement:

SELECT EMPNO, ACSTDATE FROM CORPDATA.PROJACT INNER JOIN CORPDATA.EMPPROJACT USING (PROJNO, ACTNO) WHERE ACSDATE > ’1982-12-31’

left outer join. LEFT OUTER keyword is required.

SELECT EMPNO, LASTNAME, PROJNO FROM CORPDATA.EMPLOYEE LEFT OUTER JOIN CORPDATA.PROJECT ON EMPNO = RESPEMP WHERE LASTNAME > ’S’

The result of this query contains some employees that do not have a project number.

UNION

The UNION clubs the output of both the tables/files. Duplicates are displayed only once

For example, look at the UNION clause in this statement:

If you want to keep duplicates in the result of a UNION operation, specify the UNION ALL keyword instead of just UNION

Using Subqueries

* You can use subqueries in a search condition as another way to select data. Subqueries can be used anywhere an expression can be used
* You can include a subquery in a WHERE or HAVING clause by using a basic or quantified comparison, the IN keyword, or the EXISTS keyword
* Subqueries further refine your search conditions in SELECT statements.

SELECT EMPNO, LASTNAME, JOB FROM CORPDATA.EMPLOYEE WHERE EMPNO IN (SELECT EMPNO FROM CORPDATA.EMPPROJACT WHERE PROJNO = ’MA2100’)