

TEMATISKAIS PLĀNS

3. TEMATS

Kā automātiski apstrādāt un attēlot datus? (45 stundas)

Pilnveidot prasmi iegūt, apstrādāt un vizuāli attēlot datus, izmantojot programmēšanas bibliotēkas, strukturētus datu avotus un vienkāršus datu analīzes paņēmienus.

Tematā realizētie valsts pamatizglītības standarta sasniedzamie rezultāti

- T.O.2.3.2. Izmanto programmēšanas un lietotņu piedāvātās iespējas datu apstrādei, automatizācijai un darba produktivitātes paaugstināšanai, lietojot programmēšanas bibliotēkas.
- T.V.2.3.5. Apstrādā izklājlapu datus programmā, veicot nepieciešamos aprēķinus, datu atlasī un strukturēšanu atbilstoši uzdevumam.
- T.V.2.3.6. Veic datu pārstrukturēšanu un attēlošanu, izmantojot datu analīzes automatizācijas un vizualizācijas rīkus.
- T.V.2.3.7. Izmanto datubāzes un strukturētus datu avotus, veic informācijas izguvi un iegūto datu pēcāpstrādi programmā.
- T.V.2.3.8. Izmanto tīkla un interneta pakalpojumus datu ieguvei un apmaiņai, ievērojot drošības pamatprincipus.
- T.A.2.4.10. Izmanto programmēšanas valodas un bibliotēku dokumentāciju un palīdzības sistēmu datu apstrādes risinājumu izstrādē.
- T.A.2.4.11. Skaidro ar piemēriem API (programmsaskarnes) jēdzienu un izmanto API un gatavas bibliotēkas datu ieguves un analīzes uzdevumos.
- T.A.2.4.13. Izvēlas un lieto atbilstošas programmēšanas valodas konstrukcijas, datu tipus un dažādas bibliotēkas, veidojot programmas doto uzdevumu un problēmu risinājumam.
- T.A.2.4.14. Izmanto dažādas datu struktūras (t. sk. masīvi, kopas, ieraksti, steks, rinda, saraksts, koks, grafs, datne) un ar tiem saistītos pamatalgoritmus.
- T.A.2.4.17. Izveido vienkāršu datu apstrādes programmatūru (sistēmu), datu uzglabāšanai izmantojot paša veidotu datubāzi ar vairākām tabulām.
- T.A.2.4.18. Izmanto mašīnmācīšanās pamatjēdzienus un gatavus rīkus datu analīzes uzdevumu risināšanai.

3. TEMATS. Kā automātiski apstrādāt un attēlot datus?		45 stundas
1	Ievads programmēšanas bibliotēku izmantošanā	2
2	Izklājlapu datu apstrāde, izmantojot programmēšanas bibliotēkas	4
3	Teksta datu apstrāde, izmantojot strukturētus datņu formātus (CSV, JSON)	5
4	Zināšanu pārbaude Nr. 3.1	2
5	Darbs ar tīkla un datu apmaiņas bibliotēkām	6
6	Timekļa datu izguves pamatprincipi	4
7	Datu apstrādes un vizualizācijas bibliotēku izmantošana	4
8	Zināšanu pārbaude Nr. 3.2	1
9	Mašīnmācīšanās pamatjēdzieni un to pielietojums datu analīzē	5
10	Informācijas izguve un apstrāde no datubāzēm (SQL valodas pamati)	10
11	Zināšanu pārbaude Nr. 3.3	2

Atbilstība valsts vispārējās vidējās izglītības standartam

Apakštēmas Nr.	Apakštēmas nosaukums	Standarta punkti
1	Ievads programmēšanas bibliotēku izmantošanā	2.4.10, 2.4.12
2	Izklājlapu datu apstrāde, izmantojot programmēšanas bibliotēkas	2.3.5, 2.4.13
3	Teksta datu apstrāde, izmantojot strukturētus datņu formātus (CSV, JSON)	2.4.14, 2.4.13
4	Zināšanu pārbaude Nr. 3.1	—
5	Darbs ar tīkla un datu apmaiņas bibliotēkām	2.3.8, 2.4.10
6	Timekļa datu izguves pamatprincipi	2.3.8, 2.4.13
7	Datu apstrādes un vizualizācijas bibliotēku izmantošana	2.3.6, 2.4.13
8	Zināšanu pārbaude Nr. 3.2	—
9	Mašīnmācīšanās pamatjēdzieni un to pielietojums datu analizē	2.4.18
10	Informācijas izguve un apstrāde no datubāzēm (SQL valodas pamati)	2.3.7, 2.4.17
11	Zināšanu pārbaude Nr. 3.3	—

Sasniedzamie rezultāti

Zināšanas

- Dati var tikt glabāti un apstrādāti dažādos formātos, piemēram, teksta datnēs, strukturētos datņu formātos (CSV, JSON), izklājlapās un datubāzēs.
- Programmēšanas bibliotēkas nodrošina iespējas automatizēt datu nolasīšanu, apstrādi un analīzi.
- Datu validācija ir būtiska, lai nodrošinātu datu korektumu un novērstu kļūdas datu ievadē vai importēšanā.
- Datu struktūras ļauj sistemātiski uzglabāt un apstrādāt lielu datu apjomu programmā.
- Datubāzu izmantošana un SQL vaicājumi nodrošina strukturētu datu glabāšanu un mērķtiecīgu informācijas izguvi.
- Tīmekļa resursi un API ļauj automatizēti iegūt datus no ārējiem informācijas avotiem.
- Datu vizualizācija palīdz analizēt un interpretēt apstrādātos datus un pamatot secinājumus.

Prasmes

- Raksta programmas datu nolasīšanai, validācijai un apstrādei no strukturētiem datņu formātiem (CSV, JSON). (*T.A.2.4.14., T.A.2.4.13.*)
- Izmanto programmēšanas bibliotēkas datu apstrādei, automatizācijai un darba produktivitātes paaugstināšanai. (*T.V.2.3.2., T.A.2.4.10.*)
- Veic datu atlasīšanu, filtrēšanu, pārstrukturēšanu un apkopošanu programmā. (*T.V.2.3.6., T.A.2.4.13.*)
- Izmanto SQL vaicājumu pamatus informācijas izguvei no datubāzes un iegūto datu apstrādei. (*T.V.2.3.7., T.A.2.4.17.*)
- Izmanto tīkla un datu apmaiņas bibliotēkas datu ieguvei no ārējiem avotiem. (*T.V.2.3.8.*)
- Izmanto API un gatavas bibliotēkas datu ieguves un analīzes uzdevumos. (*T.A.2.4.11.*)
- Veido datu vizualizācijas grafikos un diagrammās, izvēloties atbilstošu attēlošanas veidu. (*T.V.2.3.6.*)
- Izmanto mašīnmācīšanās pamatjēdzienus un gatavus rīkus datu analīzes uzdevumu risināšanai. (*T.A.2.4.18.*)

Komplekss sasniedzamais rezultāts

- Izstrādā programmu, kas automātiski iegūst datus no datnēm, tīmekļa resursiem vai datubāzes, veic datu apstrādi un attēlo rezultātus vizuālā formā, izmantojot piemērotas datu struktūras, programmēšanas bibliotēkas un SQL vaicājumus. (*T.O.2.3.2., T.V.2.3.6., T.V.2.3.7., T.A.2.4.14., T.A.2.4.17.*)
- Programmas izstrādes procesā analizē datu avotus, izvērtē iegūto rezultātu korektumu un skaidro izmantoto risinājumu. (*T.A.2.4.10.*)

Ierādumi

- Attīsta ieradumu sistemātiski organizēt, pārbaudīt un analizēt datus, izvērtējot to ticamību un kvalitāti. (*Tikums – gudrība; vērtība – darbs*)
- Attīsta ieradumu izmantot programmēšanu un automatizāciju ikdienas uzdevumu risināšanai un datiem pamatotu lēmumu pieņemšanai. (*Tikums – atbildība; vērtība – darbs*)
- Attīsta ieradumu atbildīgi, ētiski un droši izmantot digitālos rīkus, datu avotus un automatizācijas risinājumus. (*Tikums – atbildība; vērtība – drošība*)

Temata saturs un tā realizācija

Temata apakšdaļa	Saturs
Ievads programmēšanas bibliotēku izmantošanā	Skolēni apgūst <i>Python</i> bibliotēku un moduļu izmantošanas pamatprincipus, veidojot izpratni par to, kā programmēšanā atkārtoti tiek izmantoti jau gatavi risinājumi un kā pašiem strukturēt kodu loģiski nodalītās daļās. Tēmas ietvaros skolēni iepazīstas gan ar standarta bibliotēkām, gan ar paša veidotām bibliotēkām, sasaistot programmēšanu ar reāliem uzdevumiem matemātikā un fizikā. Aktualizācijas posmā skolēni analizē, vai programmētājs katru reizi raksta visu no jauna, vai arī izmanto jau gatavas funkcijas un rīkus. Balstoties uz ikdienas piemēriem, skolēni nonāk pie secinājuma, ka bibliotēkas ļauj būtiski ietaupīt laiku un samazināt kļūdu iespējamību, vienlaikus padarot programmu pārskatāmāku. Apjēgšanas posmā skolēni iepazīstas ar bibliotēkas jēdzienu programmēšanā, <i>import</i> sintaksi un dokumentācijas nozīmi. Skolēni apgūst, kā pieslēgt un izmantot standarta bibliotēku <i>math</i>, kā arī izprot atšķirību starp iebūvētām funkcijām un paša veidotu moduli. Tiek veidota izpratne par failu struktūru projektā un par to, kā bibliotēka darbojas kā atkārtoti izmantojama funkciju kopums. Praktiskajā darbībā skolēni izveido savu vienkāršu bibliotēku ar algu aprēķina funkciju, atdalot aprēķinu loģiku no galvenās programmas. Otrajā stundā skolēni padziļina izpratni, pielietojot bibliotēku fizikas kontekstā – vektoru attēlošanai, saskaitīšanai un reizināšanai ar skaitli. Skolēni izmanto izveidoto bibliotēku, lai risinātu vienkāršus pārvietojuma un spēku uzdevumus, tādējādi sasaistot programmēšanu ar citu mācību priekšmetu saturu. Refleksijas posmā skolēni izvērtē, kā bibliotēku izmantošana palīdz strukturēt programmu, samazina koda dublēšanos un atvieglo uzdevumu risināšanu. Skolēni secina, ka viena un tā paša risinājuma ievietošana bibliotēkā ļauj to izmantot dažādos kontekstos – gan matemātikā, gan fizikā, gan citās tēmās, veidojot pamatu sistemātiskai un ilgtspējīgai programmēšanas praksei.
Izkājlapu datu apstrāde, izmantojot programmēšanas bibliotēkas	Skolēni apgūst izklājlapu datu apstrādi ar programmēšanas bibliotēkām, veidojot izpratni, ka <i>Excel</i> tabula var kalpot kā strukturētu datu avots automatizētiem aprēķiniem, pārbaudēm un atskaišu sagatavošanai. Tēmas ietvaros skolēni apgūst bibliotēku <i>openpyxl</i> kā rīku darbam ar <i>.xlsx</i> failiem. Aktualizācijas posmā skolēni secina, ka programmēšana ļauj automatizēt atkārtotas darbības: nolasīt datus, aprēķināt vērtības pēc noteikta algoritma un saglabāt rezultātu strukturētā veidā. Apjēgšanas posmā skolēni apgūst pamatjēdzienus un darbības principu: darbagrāmata <i>workbook</i>, darblapa <i>worksheet</i>, šūnas adrese A2, piekļuve šūnai ar <i>ws["A2"].value</i> un <i>ws.cell(row=..., column=...).value</i>. Skolēni apgūst, kā ar <i>load_workbook()</i> atvērt failu, kā izvēlēties darblapu pēc nosaukuma un kā korekti pabeigt darbu ar failu. Tiek veidota izpratne par cikla izmantošanu, lai apstrādātu vairākas rindas, kā arī par rezultātu saglabāšanu jaunā failā, lai sākotnējie dati paliktu nemainīti. Praktiskajās darbībās skolēni pakāpeniski apgūst darbu ar <i>Excel</i> datnēm, izmantojot <i>Python</i> bibliotēku <i>openpyxl</i>. Skolēni nolasa datus no konkrētām <i>Excel</i> darblapas šūnām un izmanto tos vienkāršiem aprēķiniem. Tālāk risinājums tiek vispārināts, izmantojot ciklu, lai apstrādātu vairākas rindas un veiktu aprēķinus visiem ierakstiem. Skolēni ieraksta aprēķinātos rezultātus atpakaļ <i>Excel</i> darblapā. Turpinājumā tiek apgūta tabulas

	noformēšana, izmantojot šūnu stilus un formatēšanu. Noslēgumā skolēni apstrādā vairākas darblapas vienā darbagrāmatā un izveido vienkāršu kopsavilkuma tabulu. Refleksijas posmā skolēni analizē izveidoto programmas risinājumu datu apstrādei <i>Excel</i> vidē. Skolēni pārbauda, vai programma korekti nolasa datus no darblapas, veic aprēķinus un ieraksta rezultātus tabulā. Tiek izvērtēta cikla izmantošanas priekšrocība datu apstrādē vairākām rindām, kā arī analizēts izveidotās tabulas noformējums un kopsavilkuma rezultātu pareizība. Skolēni formulē secinājumus par izmantotā risinājuma efektivitāti un iespējamiem uzlabojumiem.
Teksta datu apstrāde, izmantojot strukturētus datņu formātus (CSV, JSON)	Skolēni apgūst teksta datu apstrādi, izmantojot strukturētus datņu formātus (<i>CSV</i>, <i>JSON</i>), veidojot izpratni par to, kā neapstrādāts teksts un strukturēti teksta faili var tikt izmantoti datu saglabāšanai. Tēmas ietvaros skolēni pāriet no brīvā teksta analīzes uz strukturētu datu lasīšanu, apstrādi un rakstīšanu, izmantojot <i>Python</i> standarta bibliotēkas <i>re</i>, <i>csv</i> un <i>json</i>. Aktualizācijas posmā skolēni analizē teksta datu piemērus (piemēram, kontaktu sarakstu, mērījumu rezultātus vai pieteikumu datus teksta formā) un identificē tajos atkārtojošus informācijas elementus. Skolēni izvērtē, kā šādi dati tiek strukturēti un saglabāti <i>CSV</i> un <i>JSON</i> datnēs, un secina, ka datu automātiska apstrāde ar programmu ļauj efektīvāk atrast, pārbaudīt un saglabāt informāciju. Apjēgšanas posmā skolēni apgūst teksta datu apstrādi <i>Python</i> valodā, izmantojot regulārās izteiksmes un strukturētus datņu formātus. Skolēni iepazīstas ar <i>CSV</i> un <i>JSON</i> datņu struktūru un apgūst datu lasīšanu un saglabāšanu, izmantojot <i>Python</i> bibliotēkas <i>re</i>, <i>csv</i> un <i>json</i>. Praktiskajās darbībās skolēni izpilda uzdevumus, kuros tekstā tiek meklēti un pārbaudīti datu fragmenti, iegūtā informācija tiek saglabāta <i>CSV</i> datnē un apstrādātie dati tiek saglabāti <i>JSON</i> formātā. Refleksijas posmā skolēni analizē izveidotos risinājumus un salīdzina <i>CSV</i> un <i>JSON</i> datņu izmantošanas iespējas datu saglabāšanai.
Darbs ar tīkla un datu apmaiņas bibliotēkām	Skolēni apgūst darbu ar tīkla un datu apmaiņas bibliotēkām, veidojot izpratni par to, kā ierīces apmainās ar datiem tīklā un kā šo procesu var novērot, pārbaudīt un realizēt programmā. Tēmas ietvaros skolēni sasaista tīkla pamatzināšanas (IP adreses, protokoli, porti, tīkla ierīces) ar praktisku datu apmaiņas uzdevumu risināšanu <i>Python</i> vidē, izmantojot <i>socket</i> bibliotēku. Aktualizācijas posmā skolēni analizē situācijas, kurās nepieciešams noteikt tīkla parametrus un diagnosticēt savienojumu (Wi-Fi maiņa, piekļuve tīkla ierīcēm, savienojuma stabilitāte). Skolēni secina, ka tīkla darbību var pārbaudīt gan ar sistēmas rīkiem (<i>ipconfig</i>, <i>ping</i>), gan ar programmēšanas paņēmieniem, kas ļauj automatizēt tīkla pārbaudes un uzraudzību. Apjēgšanas posmā skolēni nostiprina izpratni par tīkla veidiem un ierīcēm, lokālās un publiskās IP adreses atšķirībām, kā arī protokolu nozīmi datu apmaiņā. Tālāk skolēni apgūst tīkla portu jēdzienu, izprot saistību starp IP adresi, portu un pakalpojumu, kā arī salīdzina transporta protokolus <i>TCP</i> un <i>UDP</i>. Paralēli tiek veidota izpratne par klienta–servera arhitektūru un komunikācijas plūsmu: savienojuma izveide, datu nosūtīšana, saņemšana un savienojuma aizvēršana. Praktiskajās darbībās skolēni izstrādā risinājumus, kuros tīkla parametri tiek nolasīti, salīdzināti un izmantoti programmas darbībā: tiek noteikta IP adrese un fiksētas tās izmaiņas, pārbaudīta portu pieejamība un izdarīti secinājumi par drošības aspektiem, kā arī izveidots vienkāršs klienta–servera saziņas prototips.

	Refleksijas posmā skolēni izvērtē, kā tīkla parametri ietekmē programmas darbību. Skolēni formulē secinājumus par tīkla bibliotēku nozīmi datu apmaiņas risinājumos un to, kā klienta–servera modelis tiek izmantots reālos digitālos pakalpojumos, veidojot pamatu turpmākiem tīmekļa un API datu apstrādes uzdevumiem.
Tīmekļa datu izguves pamatprincipi	Skolēni apgūst tīmekļa datu izguves pamatprincipus, veidojot izpratni par to, kā programmā iegūt publiski pieejamu informāciju no tīmekļa vietnēm, kā atšķiras <i>HTML</i> saturs no strukturētiem datu formātiem un kā iegūtos datus atlasīt, pārbaudīt un saglabāt turpmākai izmantošanai. Tēmas ietvaros skolēni iepazīstas ar <i>URL</i> adresi, <i>HTTP</i> pieprasījumu un atbildi, statusa kodiem, kā arī atbildīgas tīmekļa datu izguves principiem. Aktualizācijas posmā skolēni analizē situācijas, kurās informācija no tīmekļa vietnēm var būt noderīga turpmākai apstrādei, piemēram, ziņu virsrakstu, cenu, citātu vai kontaktinformācijas apkopošanā. Skolēni secina, ka manuāla šādu datu atlase ir laikietilpīga, tādēļ datu izguvei var izmantot programmu. Apjēgšanas posmā skolēni iepazīstas ar tīmekļa datu izguves procesu. Skolēni apgūst, kā ar bibliotēku <i>requests</i> iegūt lapas saturu un kā ar bibliotēku <i>BeautifulSoup</i> atlasīt vajadzīgos <i>HTML</i> elementus. Skolēni veido izpratni par to, kā noteikt, kuros elementos atrodas vajadzīgā informācija, kā nolasīt tekstu un atribūtus, kā arī kā sagatavot iegūtos datus saglabāšanai. Praktiskajās aktivitātēs skolēni izstrādā programmas, kurās tiek iegūts tīmekļa lapas saturs, atlasīti vajadzīgie dati, veikta to filtrēšana pēc nosacījuma un saglabāšana strukturētā formātā. Skolēni izmanto <i>HTML</i> elementu meklēšanu pēc taga un klases, nolasā tekstu un saites adresi, pārbauda iegūto datu korektumu un saglabā rezultātus <i>JSON</i> vai <i>CSV</i> datnē. Refleksijas posmā skolēni analizē izveidoto risinājumu un izvērtē, kā tika atrasti vajadzīgie <i>HTML</i> elementi, kā notika datu atlase un kāpēc iegūtos rezultātus ir lietderīgi saglabāt strukturētā formātā. Skolēni formulē secinājumus par tīmekļa datu izguves iespējām, ierobežojumiem un nozīmi datu apstrādē.
Datu apstrādes un vizualizācijas bibliotēku izmantošana	Skolēni apgūst datu apstrādi un vizualizāciju izmantojot <i>Python</i> bibliotēkas <i>statistics</i>, <i>pandas</i> un <i>matplotlib</i>. Tēmas ietvaros skolēni veido izpratni par to, kā no datiem iegūt pārskatāmu informāciju, aprēķināt vienkāršus statistiskos rādītājus, atlasīt vajadzīgos datus un attēlot tos grafikā vai diagrammā. Aktualizācijas posmā skolēni analizē situācijas, kurās lēmumu pieņemšanai nepietiek tikai ar atsevišķiem skaitļiem, bet nepieciešams apkopot, salīdzināt un pārskatāmi attēlot datus. Skolēni secina, ka lielāku datu apjomu apstrādei un analīzei ir lietderīgi izmantot <i>Python</i> bibliotēkas. Apjēgšanas posmā skolēni apgūst datu apstrādes pamatdarbības <i>Python</i> vidē. Skolēni aprēķina vidējo vērtību, mediānu, minimumu un maksimumu, izmantojot bibliotēku <i>statistics</i>. Tālāk skolēni iepazīstas ar bibliotēku <i>pandas</i>, ielādē datus no <i>CSV</i> datnes, apskata tabulas saturu, atlasa rindas pēc nosacījuma un iegūst datu kopsavilkumu. Turpinājumā skolēni apgūst datu vizualizācijas pamatprincipus, veidojot līniju grafikus un stabiņu diagrammas. Praktiskajās darbībās skolēni izpilda uzdevumus, kuros ielādē datus no <i>CSV</i> datnes, aprēķina vienkāršus statistiskos rādītājus, analizē datu kopu un attēlo rezultātus grafikā vai diagrammā. Skolēni sagatavo datus vizualizācijai, izvēlas piemērotu attēlošanas veidu, pievieno nosaukumu un asu apzīmējumus, kā arī saglabā iegūto vizualizāciju failā. Refleksijas posmā skolēni analizē iegūtos rezultātus un izvērtē, kā datu apstrāde un vizualizācija palīdz saskatīt tendences, salīdzināt vērtības un formulēt secinājumus. Skolēni pamato, kāpēc konkrētā situācijā izvēlēts noteikts datu attēlošanas veids un ko no iegūtajiem datiem var secināt.

Mašīnmācīšanās pamatjēdzieni un to pielietojums datu analīzē	Skolēni apgūst mašīnmācīšanās pamatjēdzienus un to izmantošanu datu analīzē, veidojot izpratni par to, kā no datiem var iegūt prognozes un pieņemt lēmumus. Tēmas ietvaros skolēni iepazīstas ar pazīmju un mērķa mainīgā jēdzieniem, datu sadalīšanu apmācības un testa kopās, kā arī vienkārša mašīnmācīšanās modeļa darbības principu. Aktualizācijas posmā skolēni analizē situācijas, kurās lēmumu pieņemšanai nepietiek ar iepriekš uzrakstītiem noteikumiem, bet nepieciešams izmantot datus un līdzīgus piemērus. Skolēni secina, ka datu analīzē iespējams izmantot mašīnmācīšanās pieeju. Apjēgšanas posmā skolēni apgūst atšķirību starp datiem, informāciju un zināšanām, salīdzina nosacījuma loģiku ar mašīnmācīšanās pieeju un iepazīstas ar mašīnmācīšanās modeļa darbības pamatprincipiem. Skolēni apgūst, kas ir pazīmes, mērķa mainīgais, apmācības dati un testa dati, kā arī iepazīstas ar K tuvāko kaimiņu algoritmu kā piemēru lēmuma pieņemšanai pēc līdzīgiem datiem. Praktiskajās darbībās skolēni strādā ar vienkāršu datu kopu, ielādē datus no CSV datnes, definē pazīmes un mērķa mainīgo, sadala datus apmācības un testa kopās, apmāca vienkāršu mašīnmācīšanās modeli un veic prognozi jaunam piemēram. Skolēni analizē iegūto rezultātu un salīdzina to ar lēmumu, kas tiktu pieņemts, izmantojot tikai nosacījumus. Refleksijas posmā skolēni izvērtē, kā dati ietekmē modeļa prognozi, kāpēc nepieciešams testēt modeli ar iepriekš neredzētiem datiem un kādos gadījumos mašīnmācīšanās pieeja var būt piemērotāka par iepriekš definētiem noteikumiem.
Informācijas izguve un apstrāde no datubāzēm	Skolēni apgūst informācijas izguvi un apstrādi no datubāzēm, veidojot izpratni par strukturētu datu glabāšanu, atlasīšanu un izmantošanu praktisku uzdevumu risināšanā. Tēmas ietvaros skolēni iepazīstas ar datubāzes, tabulas, lauka, ieraksta, primārās un ārējās atslēgas jēdzieniem, kā arī apgūst <i>SQL</i> vaicājumu izmantošanu datu atlasei, filtrēšanai, kārtošānai, apvienošanai un apkopošanai. Aktualizācijas posmā skolēni analizē situācijas, kurās nepieciešams glabāt un atlasīt lielu datu apjomu, piemēram, skolēnu, pasūtījumu, darījumu vai mērījumu datus. Skolēni secina, ka šādu datu glabāšanai un apstrādei ir lietderīgi izmantot datubāzes. Apjēgšanas posmā skolēni apgūst datubāzes struktūras pamatjēdzienus un iepazīstas ar <i>SQL</i> valodas iespējām. Skolēni veido un analizē vaicājumus, izmantojot <i>SELECT</i>, <i>WHERE</i>, <i>ORDER BY</i>, <i>JOIN</i>, <i>GROUP BY</i> un agregātfunkcijas. Tālāk skolēni iepazīstas ar <i>SQL</i> un <i>Python</i> integrāciju, izpildot <i>SQL</i> vaicājumus no <i>Python</i> programmas un apstrādājot iegūtos rezultātus. Praktiskajās darbībās skolēni izpilda uzdevumus, kuros atlasa datus no vienas vai vairākām tabulām, apvieno saistītus datus, veic aprēķinus un sagatavo vienkāršus pārskatus. Skolēni izmanto <i>SQL</i> vaicājumus datu filtrēšanai, kārtošānai, grupēšanai un apkopošanai, kā arī ielādē rezultātus <i>Python</i> vidē turpmākai apstrādei. Refleksijas posmā skolēni analizē izveidotos vaicājumus un izvērtē, kā datubāzu izmantošana palīdz efektīvi iegūt, apstrādāt un interpretēt informāciju. Skolēni formulē secinājumus par <i>SQL</i> vaicājumu precizitāti, datu kvalitātes nozīmi un datubāzu izmantošanas iespējām praktisku problēmu risināšanā.